ANSI X3.230-1994

American National Standard

for Information Technology – Fibre Channel – Physical and Signaling Interface (FC-PH)

ANSI X3.230-1994

ANSI American National Standards Institute
11 West 42nd Street
New York, New York
10036

ANSI®
X3.230-1994

American National Standard
for Information Technology –

Fibre Channel – Physical and Signaling Interface (FC-PH)

Secretariat

Information Technology Industry Council

Approved November 14, 1994

American National Standards Institute, Inc.

Abstract

This standard describes the point-to-point physical interface, transmission protocol, and signaling protocol of a high-performance serial link for support of the higher level protocols associated with HIPPI, IPI, SCSI, IP, and others.

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirem nts for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made toward their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

CAUTION: The developers of this standard have requested that holders of patents that may be required for the implementation of the standard disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard.

As of the date of publication of this standard, following calls for the identification of patents that may be required for the implementation of the standard, notice of one or more such claims has been received.

By publication of this standard, no position is taken with respect to the validity of this claim or of any rights in connection therewith. The known patent holder(s) has (have), however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the publisher.

No further patent search is conducted by the developer or publisher in respect to any standard it processes. No representation is made or implied that this is the only license that may be required to avoid infringement in the use of this standard.

Published by

American National Standards Institute
11 West 42nd Street, New York, New York 10036

Printed in the United States of America

Contents

Tables

Figures

For word (This foreword is not part of American National Standard X3.230-1994.)

This standard describes the point-to-point physical interface, transmission protocol, and signaling protocol of a high-performance serial link for support of the higher-level protocols associated with HIPPI, IPI, SCSI, and others.

This standard was developed by Task Group X3T9.3 of Accredited Standards Committee X3 during 1990. The standards approval process started in 1992. This document includes 23 annexes, which are informative and are not considered part of the standard.

Requests for interpretation, suggestions for improvement or addenda, or defect reports are welcome. They should be sent to the X3 Secretariat, Information Technology Industry Council, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Information Technology, X3. Committee approval of the standard does not necessarily imply that all committee members voted for approval. At the time it approved this standard, the X3 Committee had the following members:

James D. Converse, Chair
Donald C. Loughry, Vice-Chair
Joanne M. Flanagan, Secretary

Organization Represented	*Name of Representative*
American Nuclear Society	Geraldine C. Main
	Sally Hartzell (Alt.)
AMP, Inc.	Edward Kelly
	Charles Brill (Alt.)
Apple Computer, Inc.	David K. Michael
AT&T Global Information Systems	Robert K. Kramer
	Thomas F. Frost (Alt.)
Bull HN Information Systems, Inc.	William M. George, Jr.
Compaq Computers	Ed Olkkola
Digital Equipment Corporation	Scott K. Jameson
	Richard Hovey (Alt.)
Eastman Kodak Company	James D. Converse
	Michael Nier (Alt.)
Guide International, Inc.	Frank Kirshenbaum
	Tony Gualtieri (Alt.)
Hewlett-Packard	Donald C. Loughry
Hitachi America, Ltd.	John Neumann
	Kei Yamashita (Alt.)
Hughes Aircraft Company	Harold Zebrack
IBM Corporation	Joel Urman
	Mary Anne Lawler (Alt.)
Institute for Certification of Computer Professionals (ICCP)	Kenneth Zemrowski
National Communications Systems	Dennis Bodson
	Granger Kelley (Alt.)
National Institute of Standards and Technology	Michael D. Hogan
	James H. Burrows (Alt.)
Neville & Associates	Carlton Neville
Northern Telecom, Inc.	Mel Woinsky
	Subhash Patel (Alt.)
Recognition Technology Users Association	Herbert P. Schantz
	Gerald Farm r (Alt.)
Share, Inc.	Gary Ainsworth
	David Th wlis (Alt.)

Organization Represented	*Name of Representative*
Sony Corporation of America	Michael Deese
Storage Technology Corporation	Joseph S. Zajaczkowski
	Samuel D. Cheatham (Alt.)
Sun Microsystems, Inc.	Gary S. Robinson
Sybase, Inc.	Donald R. Deutsch
Texas Instruments, Inc.	Clyde Camp
	Fritz Whittington (Alt.)
3M Company	Eddie T. Morioka
	Paul D. Jahnke (Alt.)
Unisys Corporation	John L. Hill
	Stephen P. Oksala (Alt.)
U.S. Department of Defense	William Rinehuls
	C. J. Pasquariello (Alt.)
U.S. Department of Energy	Alton Cox
	Neil S. Jarrett (Alt.)
U.S. General Services Administration	Patrick Plunkett
	Douglas K. Arai (Alt.)
Wintergreen Information Services	Jack L. Wheeler
Xerox Corporation	Roy Pierce
	William Ted Smith (Alt.)

Subcommittee X3T9 on Computer Input/Output Interfaces, which reviewed this standard, had the following members:

Delbert L. Shoemaker, Chair
Robert L. Fink, Vice-Chair

Charles Brill
William Burr
Jefferson Connell
Stephen Cooper
Roger Cummings
James Hamstra
David Husak
Reinhard Knerr
Lawrence Lamers
George Michael
Arthur Miller
Gene Milligan

Doug Morrissey
Steve Myers
Roger Pandanda
Everett Rigsbee
Floyd Ross
Amit Shah
Geoffrey Stone
Pat Thaler
Don Tolmie
Schelto van Doorn
Jim Vogt
Carl Zeitler, Jr.

Task Group X3T9.3 on Device Level Interfaces, which developed this standard, had the following participants:

Roger Cummings, Chair
Carl Zeitler, Vice-Chair

J. Albano
D. Allan
R. Anderson
T. Anderson
K. Annamalai
C. Barron
B. Beach
A. Bedi
B. Bellino
H. Brandt
C. Brill
R. Brown
M. Carboni
B. Carlson
K. Chan
K. Chennappan
B. Cook
J. Connell
B. Cornelius
S. Dean
D. Deel
J. Dudgeon
R. Dugan
S. Erler
G. Frazier
G. Frederickson
E. Frymoyer
B. Gallagher
S. Gillett
E. Grivna
J. Guedj
D. Guss
D. Gustafson
M. Halvorson
D. Hansen
K. Hardwick
M. Hartmann
M. Henske
D. Hepner
J. Huber
J. Hughes
R. Johnson
T. Kaminski
J. Kerr
R. Knerr
F. Koperda
B. Kopsaftis
O. Kornblum
G. Koskowich
J. Kubinec
P. Lambert
P. Levin
R. Liebow
J. Luttrull
S. Magee
K. Malavalli
C. Martin
R. Masterson
J. Mathis
T. McClendon
G. Milligan
G. Michael
M. Miller
B. Morris
J. Morris
D. Morrissey
R. Niemi
G. Nordstrom
M. O'Donnell
K. Ocheltree
T. Palkert
E. Parish
N. Perdue
G. Phillips
R. Phillips
J. Radcliffe
P. Ramsey
G. Rara
E. Reid
J. Renwick
W. Rickard
W. Roiger
R. Ronald
G. Rouse
P. Rupert
T. Russell
M. Sachs
W. Sanderson
P. Savage
J. Scherfey
P. Schreck
J. Schuessler
P. Scott
F. Shott
K. Sivakumar
L. Sloan
J. Smith
R. Snively
R. Soderstrom
D. Somes
J. Soriano
G. Stephens
R. Southard
T. Sprenkle
G. Stager
D. Stigliani
A. Stone
B. Stoevhase
T. Szostak
R. Taborek
R. Thomsen
H. Todd
D. Tolmie
C. Towndrow
H. Truestedt
E. Vanderbeck
S. van Doorn
L. Veil
M. Vogel
P. Walford
Y. Wang
L. Whitfield
A. Widmer
J. Young
S. Zanowick

The designated editors of this document were:

Brent Carlson
KC Chennappan
Ken Jackson
Frank Shott
Joe Mathis
Jerry Radcliffe
Jerry Rouse
Jonathan Thatcher
Shelto van Doorn

Intr duction

The Fibre Channel provides a general transport vehicle for Upper Level Protocols (ULPs) such as Intelligent Peripheral Interface (IPI) and Small Computer System Interface (SCSI) command sets, the High-Performance Parallel Interface (HIPPI) data framing, IP (Internet Protocol), IEEE 802.2, and others. Proprietary and other command sets may also use and share the Fibre Channel, but such use is not defined as part of the Fibre Channel standard. Other usages such as local area network protocols and backbone configuration have been considered.

The Fibre Channel standard is organized in the following levels (see figure 2):

- *FC-0* defines the physical portions of the Fibre Channel including the fibre, connectors, and optical and electrical parameters for a variety of data rates and physical media. Coax and twisted pair versions are defined for limited distance applications. FC-0 provides the point-to-point physical portion of the Fibre Channel. A variety of physical media is supported to address variations in cable plants.

- *FC-1* defines the transmission protocol which includes the serial encoding, decoding, and error control.

- *FC-2* defines the signaling protocol which includes the frame structure and byte sequences.

- *FC-3* defines a set of services which are common across multiple ports of a node.

- *FC-4* is the highest level in the Fibre Channel standards set. It defines the mapping, between the lower levels of the Fibre Channel and the IPI and SCSI command sets, the HIPPI data framing, IP, and other Upper Level Protocols (ULPs).

Of these levels, FC-0, FC-1, and FC-2 are integrated into this FC-PH document. The Fibre Channel protocol provides a range of implementation possibilities extending from minimum cost to maximum performance. The transmission medium is isolated from the control protocol so that each implementation may use a technology best suited to the environment of use.

Figure 1 shows the relationship of this American National Standard (the highlighted rectangle) with other Fibre Channel documents. FC-EP specifies the enhanced functions added to FC-PH. FC-FG, FC-XS, and FC-DF are documents related to Fabric requirements. FC-AL specifies the arbitrated loop topology. FC-IG provides some implementation guidance. FC-SB, FC-FP, IPI-3 Disk, IPI-3 Tape, FC-LE, SCSI-FCP, SCSI-GPP, and FC-ATM are FC-4 documents.

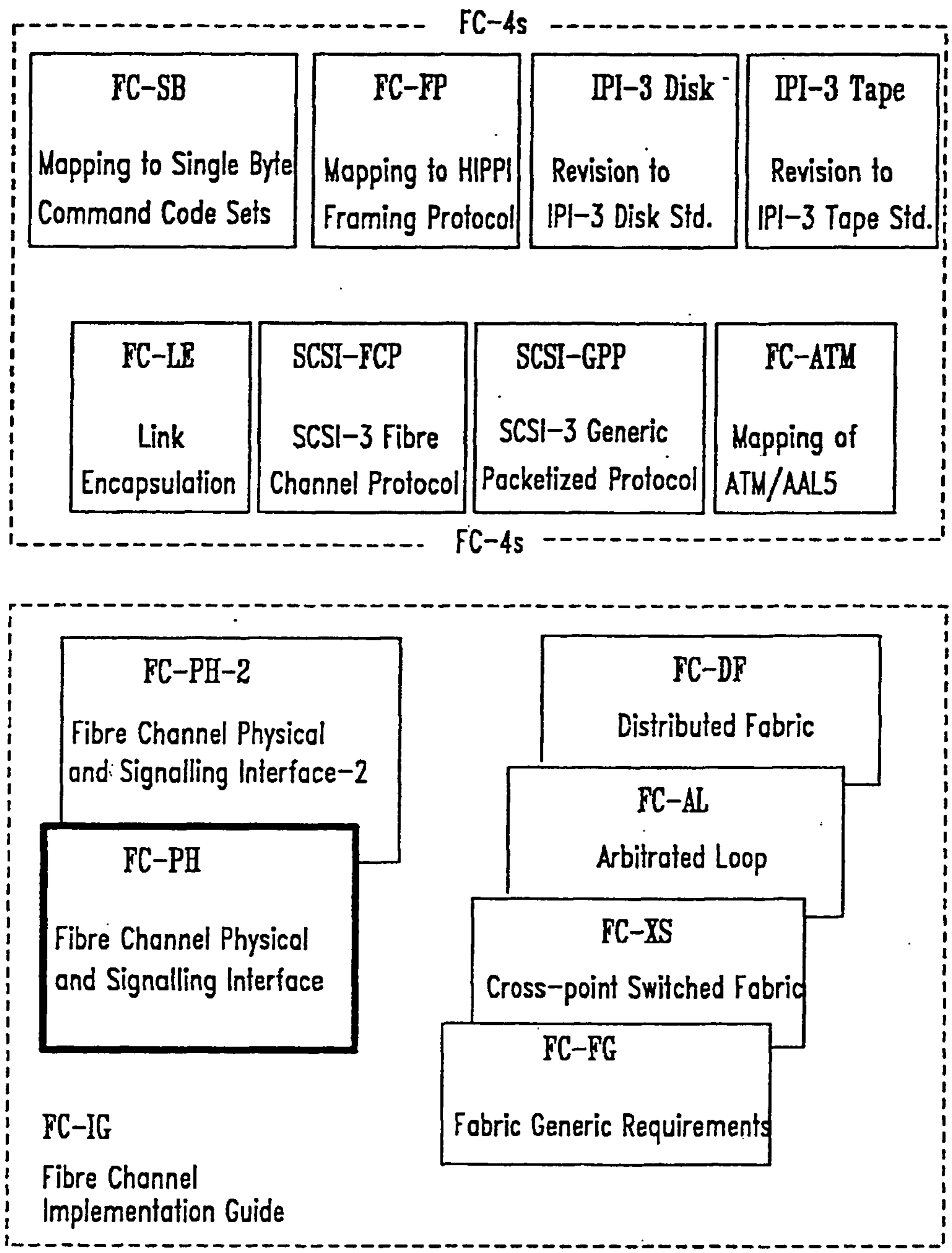


Figure 1 – Document relationship

American National Standard for Information Technology – Fibre Channel – Physical and Signaling Interface (FC-PH)

1 Scope

This standard describes the physical and signalling interface of a high performance serial link for support of the Upper Level Protocols (ULPs) associated with HIPPI, IPI, SCSI, IP and others.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the following list of standards. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 9314-2:1989, *Fiber Distributed Data Interface - Media Access Control (FDDI-MAC)*

ANSI/IEEE Std 802 - 1990, *Local and Metropolitan Area Networks: Overview and Architecture* (formerly known as IEEE Std 802.1A, Project 802: Local and Metropolitan Area Networks Standard – Overview and Architecture)

FOTP-6 (ANSI/EIA/TIA-455-6B-1992) - [1], [2] *Cable Retention Test Procedure for Fiber Optic Cable Interconnecting (March 1992)*

FOTP-29 (ANSI/EIA/TIA-455-29A-1989) - *Refractive Index Profile, Transverse Interference Method: 1st Ed. Aug. 1981, 2nd Ed. Oct. 1989.* (Measures core diameter, numerical aperture, and refractive index profile of multimode fiber) Reaffirmed 04/01/91 until 10/94

FOTP-30 (ANSI/EIA/TIA-455-30B-1991) - *Frequency Domain Measurement of Multimode Optical Fiber Information Transmission Capacity: 1st Ed. Sept. 1982, 2nd Ed. Aug. 1988, 3rd Ed. Oct. 1991*

FOTP-34 (ANSI/EIA/TIA-455-34/1985) - *Interconnection Device Insertion Loss Test, May 1985*

FOTP-44 (ANSI/EIA/TIA-455-44A/1992) - *Refractive Index Profile, Refracted Ray Method: 1st Ed. Jan. 1984, 2nd Ed. Oct. 1989, 3rd Ed. Sept. 1992.* (Measures core diameter, numerical aperture, and refractive index profile of multimode fiber)

FOTP-45 (ANSI/EIA/TIA-455-45B-1992) - *Method for Measuring Optical Fiber Geometry Using a Laboratory Microscope: 1st Ed. Sept. 1984, 2nd Ed. Aug. 1988, 3rd Ed. June 1992*

FOTP-47 (ANSI/EIA/TIA-455-47B-1992) - *Output Farfield Radiation Pattern Measurement: 1st Ed. Sept. 1983, 2nd Ed. May 1989, 3rd Ed. Aug. 1992* (Measures numerical aperture of multimode fiber)

---

1 All FOTP-xx are EIA/TIA-455-xxx and all OFSTP-xx are EIA/TIA-526-xxx. All FOTP and OFSTP references are as of 12/16/92. Note that some are listed as EIA-zzz-xx and some as EIA/TIA-zzz-xx. The reason for this is related to timing of the document development and/or revision since TIA has become an accredited organization.

2 Fiber Optic Test Procedure (FOTP) and Optical Fiber System Test Practice (OFSTP) standards are developed and published by the Electronics Industries Association under the EIA/TIA-455 and the EIA/TIA-526 series of standards. Copies may be obtained by contacting the American National Standards Institute, 11 West 42nd Street, New York, New York 10036.

FOTP-48 (ANSI/EIA/TIA-455-48B-1990) - *Diameter Measurement of Optical Fibers Using Laser-Based Measurement Instruments: 1st Ed. Dec. 1983, 2nd Ed. Oct. 1987, 3rd Ed. Dec. 1990*

FOTP-51 (ANSI/EIA/TIA-455-51A-1991) - *Pulse Distortion Measurement of Multimode Glass Optical fiber Information Transmission Capacity: 1st Ed. Sept. 1983, 2nd Ed. May 1991*

FOTP-54 (ANSI/EIA/TIA-455-54A-1990) - *Mode Scrambler Requirements for Overfilled Launching Conditions to Multimode Fibers: 1st Ed. Sept. 1982, 2nd Ed. Nov. 1990*

FOTP-58 (ANSI/EIA/TIA-455-58A-1990) - *FOTP-58 Core Diameter Measurement of Graded-Index Optical Fibers, Nov. 1990*

FOTP-107 (ANSI/EIA/TIA-107-1989) - *Return Loss for Fiber Optic Components (February 1989)*

FOTP-127 (ANSI/EIA/TIA-455-127-1991) - *Spectral Characteristics of Multimode Laser Diodes Performance, Nov. 1991*

FOTP-168 (ANSI/EIA/TIA-455-168A-1992) - *Chromatic Dispersion Measurement of Multimode Graded-Index and Single-Mode Optical Fibers by Spectral Group Delay Measurement in the Time Domain: 1st Ed. July 1987, 2nd Ed. March 1992*

FOTP-171 (ANSI/EIA/TIA-455-171-1986) - *Attenuation by Substitution Measurement- For Short Length Multimode and Single-Mode Fiber Cable Assemblies, July 1986*

FOTP-176 (Unpublished - will become EIA/TIA-455-176) - *Measurement Method for Optical Fiber Geometry by Automated Grey-Scale Analysis* (Originally got through SP ballot, which closed 03/04/88. Extensive revision since that time required that it be reballoted. Limited 30-day SP ballot was not approved by ANSI, which now requires full SP reballot;submitted to TIA 11/23/92.) [3]

FOTP-177 (ANSI/EIA/TIA-455-177-1992) - *Numerical Aperture Measurement of Graded-Index Optical Fibers: 1st Ed. Nov. 1989, 2nd Ed. Aug. 1992* ("Umbrella" document, indicating factors required by FOTP-29, FOTP-44, and FOTP-47 to map to each other)

FOTP-187 (EIA/TIA-455-187-1991) - *Engagement and Separation Force Measurement of Fiber Optic Connector Sets, June 1991*

ANSI/EIA-403-A-1990 - *Precision Coaxial Connectors for CATV Applications (75 Ohms)*

ANSI/EIA/TIA-492AAAA-1989 - *Detail Specification for 62.5 μm Core Diameter/125 μm Cladding Diameter Class 1a Multimode, Graded Index Optical Waveguide Fibers*

ANSI/EIA/TIA-492BAAA - *Detail Specification for Class IVa Dispersion-Unshifted Single-Mode Optical Waveguide Fibers Used in Communications Systems* [4]

OFSTP-2 (EIA/TIA-526-2) - *Effective Transmitter Output Power Coupled into Single-Mode Fiber Optic Cable, Sept. 1992* [4]

OFSTP-3 (ANSI/EIA/TIA-526-3-1989) - *Fiber Optic Terminal Equipment Receiver Sensitivity and Maximum Receiver Input, Oct. 1989*

OFSTP-4 (EIA/TIA-526-4) - *Optical Eye Pattern Measurement Procedure (in final stage of approval)* [3]

OFSTP-7 (EIA/TIA-526-7) - *Optical Power Loss Measurement of Installed Single-Mode Fiber Cable Plant, Jan. 1993 Draft (unpublished)* [3]

OFSTP-11 (ANSI/EIA/TIA-526-11-1991) - *Measurement of Single-Reflection Power Penalty for Fiber Optic Terminal Equipment, Dec. 1991*

OFSTP-14 (ANSI/EIA/TIA-526-14-1990) - *Optical Power Loss Measurement of Installed Multimode Fiber Cable Plant, Nov. 1990*

JIS C 5973 - *FO4 Type Connectors for Optical Fiber Cards.* [5]

ANSI/EIA/TIA 568-1991 - *Commercial Building Telecommunications Wiring Standard.*

MIL-17/2 - *RG-6 Radio Frequency Cable: Flexible Coax - 75 Ohm.* [6]

---

[3] Contact the Secretariat for more recent information.

[4] Available from the Electronic Industries Association, 2500 Wilson Boulevard, Suite 300, Arlington, VA 22201

[5] Available from Japanese Standards Association, 1-24, Akasaka 4, Minato-Ku, Tokyo 107 Japan Fax: 81-33-583-8003 or Global Engineering, 15 Iverness Way East, Englewood, CO 80112-5704 Phone: (800)854-7179 or (303)792-2181 Fax: (303) 792-2192

[6] Copies of federal and military specifications, standards, and handbooks are available from the Standardization Document Order Desk, Building 4D, 700 Robbins Avenue, Philadelphia, PA 19111-5094.

MIL-17/29 - *RG-59 Radio Frequency Cable: Flexible Coax - 75 Ohm*

MIL-17/94 - *RG-179 Radio Frequency Cable: Flexible Coax - 75 Ohm*

MIL-C-24308 - *Miniature Polarized Shell Type Non-environmental Rectangular Connector with Socket Contacts for Printed Wiring Board Termination*

MIL-C-39012 - *General Specification for Radio Frequency Coaxial Connectors*

Food and Drug Administration (FDA) / Department of Health and Human Services (DHHS) Regulations 21 CFR Chapter I, Subchapter J, Part 1040.10, *Performance standards for light-emitting products*

ANSI Z136.2-1988, *Standard for the safe use of optical fibre communication systems utilizing laser diode and LED sources*

IEC 825-1984, *Radiation safety of laser products, equipment classification, requirements and user's guide*

3 Definitions and conventions

For the purposes of this standard, the following definitions, conventions, abbreviations, acronyms and symbols apply.

3.1 Definitions

3.1.1. Active: The state of Sequence Initiator until all the Data frames for the Sequence have been transmitted. The state of Sequence Recipient until all the Data frames for the Sequence have been received. (see 24.6.1).

3.1.2. address identifier: An address value used to identify source (S_ID) or destination (D_ID) of a frame.

3.1.3. alias address identifier (alias): One or more address identifiers which may be recognized by an N_Port in addition to its N_Port Identifier. An alias address identifier is Fabric unique and may be common to multiple N_Ports.

3.1.4. Arbitrated Loop topology: A configuration that allows multiple ports to be connected serially (see FC-AL document).

3.1.5. attenuation: The transmission medium power loss expressed in units of dB.

3.1.6. average power: The optical power measured using an average reading power meter when the Fibre Channel is transmitting a specified code sequence as defined in the test procedure.

3.1.7. bandwidth: Maximum effective transfer rate for a given set of physical variants such as communication model, Payload size, Fibre speed, and overhead specified by FC-PH (see 4.7 and annex M).

3.1.8. baud: The encoded bit rate per second.

3.1.9. BB_buffer: The buffer associated with buffer-to-buffer flow control.

3.1.10. Beginning Running Disparity: The Running Disparity present at a transmitter when Encoding of the Special Code associated with an Ordered Set is initiated, or at a receiver when Decoding of the Special Character associated with an Ordered Set is initiated.

3.1.11. bit error rate (BER): The statistical probability of a transmitted bit being erroneously received in a communication system. The BER is measured by counting the number of erroneous bits at the output of a receiver and dividing by the total number of bits.

3.1.12. Bit synchronization: The state in which a receiver is delivering retimed serial data at the required BER.

3.1.13. BNC: Acronym for a Bayonet-Neil-Councilman Coaxial Cable Connector. Specifications for BNC style connectors are defined in EIA/TIA 403-A and MIL-C-39012.

3.1.14. block: A upper level construct of application data related to a single Information Category and transferred within a single Sequence.

3.1.15. **buffer:** A logical construct which holds the contents of a singl fram (i.e., contents of a frame ≤ one buffer) (see Credit and clause 17).

3.1.16. **byte:** An eight-bit entity with its least significant bit denoted as bit 0 and most significant bit as bit 7. The most significant bit is shown on the left side in FC-PH, unless specifically indicated otherwise.

3.1.17. **cable plant:** All passive communications elements (e.g., optical fibre, twisted pair, or coaxial cable, connectors, splices, etc.) between a transmitter and a receiver.

3.1.18. **centre wavelength (laser):** The nominal value of the central wavelength of the operating, modulated laser. This is the wavelength (see FOTP-127) where the effective optical power resides.

3.1.19. **centre wavelength (LED):** The average of the two wavelengths measured at the half amplitude points of the power spectrum.

3.1.20. **character:** Any Transmission Character associated by FC-1 transmission code with a FC-2 data byte or special code. Transmission characters are generated and interpreted only by FC-1.

3.1.21. **circuit:** A bidirectional path within the Fabric.

3.1.22. **Classes of service:** Different types of services provided by the Fabric and used by the communicating N_Ports (see 4.9 and 22).

3.1.23. **Class 1 service:** A service which establishes a dedicated connection between communicating N_Ports (see 4.9 and 22).

3.1.24. **Class 2 service:** A service which multiplexes frames at frame boundaries to or from one or more N_Ports with acknowledgement provided. (see 4.9 and 22).

3.1.25. **Class 3 service:** A service which multiplexes frames at frame boundaries to or from one or more N_Ports without acknowledgement (see 4.9 and 22).

3.1.26. **coaxial cable:** An electrical transmission medium consisting of concentric conductors separat d by a dielectric material with the spacings and material arranged to give a specified electrical impedance.

3.1.27. **code bit:** The smallest time period used by FC-0 for transmission on the media.

3.1.28. **code balance:** The numerical sum of the 1 bits in any 10 bits in the transmitted bit stream divided by 10 (e.g., 1110100011 has a code balance of 6/10 = 60%).

3.1.29. **code violation:** An error condition that occurs when a received Transmission Character cannot be decoded to a Valid Data Byte or Special Code using the validity checking rules specified by the Transmission Code.

3.1.30. **comma:** The seven bit sequence 0011111 or 1100000 in an encoded stream.

3.1.31. **comma character:** A Special Character containing a comma.

3.1.32. **concatenation:** A logical operation that "joins together" strings of data and is represented with the symbol "||". Two or more fields are concatenated to provide a reference of uniqueness (e.g., S_ID||X_ID).

3.1.33. **Connection:** See Dedicated Connection.

3.1.34. **Connection Initiator:** The source N_Port which initiates a Class 1 Connection with a destination N_Port through a connect-request and also receives a valid response from the destination N_Port to complete the Connection establishment.

3.1.35. **Connection Recipient:** The destination N_Port which receives a Class 1 connect-request from the Connection Initiator and accepts establishment of the Connection by transmitting a valid response.

3.1.36. **Connectionless buffers:** Receive buffers participating in Connectionless service and capable of receiving Connectionless frames (see clause 26).

3.1.37. **Connectionless frames:** Frames participating in Connectionless service (i.e., Class 1 frames with **SOFc1**, Class 2, and Class 3 frames referred to individually or collectively) (see clause 26).

3.1.38. Connectionless service: Communication between two N_Ports performed without a Dedicated Connection.

3.1.39. Continuously Increasing Relative Offset: the relationship specified between Relative Offset values contained in frame (n) and frame (n+1) of an Information Category within a single Sequence. Continuously Increasing Relative Relative Offset (RO_i) for a given Information Category i is specified by the following:
$RO_i(n+1) = RO_i(n) + \text{Length of Payload}_i(n)$
where n is ≥ 0 and represents the consecutive frame count of frames for a given Information Category within a single Sequence.
$RO_i(0) = \text{Initial Relative Offset } (IRO_i)$ for the Information Category i (see Relative Offset and Random Relative Offset).

3.1.40. Credit: The maximum number of receive buffers allocated to a transmitting N_Port or F_Port. It represents the maximum number of outstanding frames which can be transmitted by that N_Port or F_Port without causing a buffer overrun condition at the receiver (see 26.3).

3.1.41. current running disparity: The Running Disparity present at a transmitter when Encoding of a Valid Data Byte or Special Code is initiated, or at a receiver when Decoding of a Transmission Character is initiated.

3.1.42. Data Character: Any Transmission Character associated by the Transmission Code with a Valid Data Byte.

3.1.43. Data frame: A frame containing information meant for FC-4/ULP or the Link application.

3.1.44. decoding: Validity checking of received Transmission Characters and generation of Valid Data Bytes and Special Codes from those characters.

3.1.45. Dedicated Connection: A communicating circuit guaranteed and retained by the Fabric for two given N_Ports.

3.1.46. delimiter: An Ordered Set used to indicate a frame boundary.

3.1.47. Destination_Identifier (D_ID): The address identifier used to indicate the targeted destination of the transmitted frame.

3.1.48. destination N_Port: The N_Port to which a frame is targeted.

3.1.49. discard policy: An error handling policy where an N_Port is able to discard Data frames received following detection of a missing frame in a Sequence (see 29.6.1.1).

3.1.50. disconnection: The process of removing a Dedicated Connection between two N_Ports.

3.1.51. disparity: The difference between the number of ones and zeros in a Transmission Character.

3.1.52. dispersion: A term used to denote pulse broadening and distortion. The two general categories of dispersion are modal dispersion, due to the difference in the propagation velocity of the propagation modes in a multimode fibre, and chromatic dispersion, due to the difference in propagation of the various spectral components of the optical source.

3.1.53. EE_buffer: The buffer associated with end-to-end flow control.

3.1.54. electrical fall time: The time interval for the falling edge of an electrical pulse to transition from its 90% amplitude level to its 10% amplitude level.

3.1.55. electrical rise time: The time interval for the rising edge of an electrical pulse to transition from its 10% amplitude level to its 90% amplitude level.

3.1.56. encoding: Generation of Transmission Characters from Valid Data Bytes and Special Codes.

3.1.57. Exchange: The basic mechanism which transfers information consisting of one or more related non-concurrent Sequences which may flow in the same or opposite directions. An Exchange may span multiple Class 1 Dedicated Connections. The Exchange is identified by an Originator Exchange_Identifier (OX_ID) and a Responder Exchange_Identifier (RX_ID) (see clause 24).

3.1.58. Exchange_Id ntifier (X_ID): A generic reference to OX_ID and RX_ID (see Exchange).

3.1.59. Exchange Status Block: A logical construct which contains th state of an Exchange.

An Originator N_Port has an Originator Exchange Status Block and the Responder N_Port has a Responder Exchange Status Block for each concurrently active Exchange (see 4.13.4.3 and 24.8.1).

3.1.60. **Exclusive Connection:** A Class 1 Dedicated Connection without Intermix (see Dedicated Connection and 22.4).

3.1.61. **extinction ratio:** The ratio (in dB) of the average optical energy in a logic one level to the average optical energy in a logic zero level measured under modulated conditions at the specified baud rate.

3.1.62. **eye opening:** The time interval across the eye, measured at the 50% normalized eye amplitude which is error free to the specified BER.

3.1.63. **F_Port:** The Link_Control_Facility within the Fabric which attaches to an N_Port through a link. An F_Port is addressable by the N_Port attached to it, with a common well-known address identifier (hex 'FFFFFE') (see 18.3, local F_Port, and remote F_Port).

3.1.64. **Fabric:** The entity which interconnects various N_Ports attached to it and is capable of routing frames by using only the D_ID information in a FC-2 frame header.

3.1.65. **Fabric_Name:** A Name_Identifier associated with a Fabric.

3.1.66. **fibre:** A general term used to cover all transmission media specified in FC-PH (see clauses 4 and 6).

3.1.67. **Fibre Channel Name:** An Name_Identifier which is Fibre Channel unique.

3.1.68. **fibre optic cable:** A jacketed optical fibre or fibres.

3.1.69. **fiber optic test procedure (FOTP):** Standards developed and published by the Electronic Industries Association (EIA) under the EIA-RS-455 series of standards.

3.1.70. **FL_Port:** An F_Port that contains Arbitrated Loop functions associated with Arbitrated Loop topology (see FC-AL document).

3.1.71. **frame:** An indivisible unit of information used by FC-2.

3.1.72. **F_Port Name:** A Name_Identifier associated with an F_Port.

3.1.73. **Frame Content:** The information contained in a frame between its Start-of-Frame and End-of-Frame delimiters, excluding the delimiters.

3.1.74. **Hunt Group:** A set of N_Ports with a common alias address identifier managed by a single node or common controlling entity. However, FC-PH does not presently specify how a Hunt Group can be realized.

3.1.75. **Idle Word (Idle):** An ordered set of four transmission characters (see clause 11) which are normally transmitted between frames. The Idle Word is also referred to as an Idle (see 11.4).

3.1.76. **ignored:** A field that is not interpreted by the receiver.

3.1.77. **infinite buffer:** A terminology to indicate that at FC-2 level, the amount of buffer available at the Sequence Recipient is unlimited. The ULP chooses the amount of buffer per Sequence based on its MTU (maximum transfer unit).

3.1.78. **Information Category:** A frame header field indicating the category to which the frame payload belongs (e.g., Solicited Data, Unsolicited Data, Solicited Control and Unsolicited Control).

3.1.79. **information transfer:** Transfer of frames whose Payload has meaning to the cooperating FC-4s.

3.1.80. **Information Unit:** An organized collection of data specified by FC-4 to be transferred as a single Sequence by FC-2.

3.1.81. **initialization:** For FC-1 level the period beginning with power on and continuing until the transmitter and receiver of that level become Operational.

3.1.82. **Initial Relative Offset:** A Relative Offset value specified at the sending end by an upper level for a giv n block or subblock and used by the sending FC-2 in the first frame of that block or subblock (see subblock, block, and Relative

Offset). Initial Relative Offset value may be zero or non-zero.

3.1.83. interface connector: An optical or electrical connector which connects the media to the Fibre Channel transmitter or receiver. The connector consists of a receptacle and a plug.

3.1.84. Intermix: A service which interleaves Class 2 and Class 3 frames on an established Class 1 Connection (see clauses 4 and 22.4).

3.1.85. intersymbol interference: The effect on a sequence of symbols in which the symbols are distorted by transmission through a limited bandwidth medium to the extent that adjacent symbols begin to interfere with each other.

3.1.86. jitter: Deviations from the ideal timing of an event which occur at high frequencies. Low frequency deviations are tracked by the clock recovery and do not directly affect the timing allocations within a bit cell. Jitter is not tracked by the clock recovery and directly affects the timing allocations in a bit cell. For FC-PH the lower cutoff frequency for jitter is defined as the bit rate divided by 2 500. Jitter is customarily subdivided into deterministic and random components.

3.1.87. jitter, deterministic (DJ): Timing distortions caused by normal circuit effects in the transmission system. Deterministic jitter is often subdivided into duty cycle distortion (DCD) caused by propagation differences between the two transitions of a signal and data dependent jitter (DDJ) caused by the interaction of the limited bandwidth of the transmission system components and the symbol sequence.

3.1.88. jitter, random (RJ): Jitter due to thermal noise which may be modeled as a Gaussian process. The peak-to-peak value of RJ is of a probabilistic nature and thus any specified value yields an associated BER.

3.1.89. laser chirp: A phenomenon in lasers where the wavelength of the emitted light changes during modulation.

3.1.90. lev l: 1. A docum nt artifice used to group related architectural functions. No specific correspondence is intended between levels and actual implementations. 2. a specific value of voltage (e.g., voltage level).

3.1.91. link: 1. Two unidirectional fibres transmitting in opposite directions and their associated transmitters and receivers.
2. The full-duplex FC-0 level association between FC-1 entities in directly attached Ports (see Port).

3.1.92. Link_Control_Facility: A link hardware facility which attaches to an end of a link and manages transmission and reception of data. It is contained within each N_Port and F_Port.

3.1.93. local F_Port: The F_Port to which an N_Port is directly attached by a link (see remote F_Port).

3.1.94. loopback: A mode of FC-1 operation in which the information passed to the FC-1 transmitter for transmission is shunted directly to the FC-1 receiver, overriding any signal detected by the receiver on its attached fibre.

3.1.95. L_Port: An N_Port or F_Port that contains Arbitrated Loop functions associated with Arbitrated Loop topology (see FC-AL document).

3.1.96. mandatory: A function which is required to be supported by a compliant implementation of FC-PH.

3.1.97. meaningful: A control field or bit shall be applicable and shall be interpreted by the receiver, wherever it is specified as meaningful. Wherever it is specified as "not meaningful", it shall be ignored (see valid).

3.1.98. mode-partition noise: Noise in a laser based optical communication system caused by the changing distribution of laser energy partitioning itself among the laser modes (or lines) on successive pulses in the data stream. The effect is a different centre wavelength for the successive pulses resulting in arrival time jitter attributable to chromatic dispersion in the fibre.

3.1.99. Name_Identifier: A 64 bit identifier, with a 60 bit value preceded with a 4 bit Network_Address_Authority_Identifier, used to identify entities in Fibre Channel such as N_Port, Node, F_Port, or Fabric (see table 43).

3.1.100. Network_Addr ss_Authority (NAA): An organization such as CCITT or IEEE which administers network addresses (see 19.3).

3.1.101. Netw rk_Address_Authority (NAA) identifier: A four bit identifier defined in FC-PH to

indicate a Network_Address_Authority (NAA) (see 19.3).

3.1.102. NL_Port: An N_Port that contains Arbitrated Loop functions associated with Arbitrated Loop topology (see FC-AL document).

3.1.103. Node: A collection of one or more N_Ports controlled by a level above FC-2.

3.1.104. Node_Name: A Name_Identifier associated with a Node.

3.1.105. non-repeating ordered set: An Ordered Set which, when issued by FC-2 to FC-1 for transmission, is to be transmitted once.

3.1.106. Not Operational: A receiver or transmitter that is not capable of receiving or transmitting an encoded bit stream respectively, based on the rules defined by FC-PH for error control. For example, FC-1 is Not Operational during Initialization.

3.1.107. N_Port: A hardware entity which includes a Link_Control_Facility. It may act as an Originator, a Responder, or both.

3.1.108. N_Port Identifier: A Fabric unique address identifier by which an N_Port is uniquely known. The identifier may be assigned by the Fabric during the initialization procedure. The identifier may also be assigned by other procedures not defined in FC-PH. The identifier is used in the S_ID and D_ID fields of a frame.

3.1.109. N_Port Name: A Name_Identifier associated with an N_Port.

3.1.110. numerical aperture: The sine of the radiation or acceptance half angle of an optical fibre, multiplied by the refractive index of the material in contact with the exit or entrance face. See FOTP-177.

3.1.111. Open: The period of time starting when a Sequence (an Exchange) is initiated until that Sequence (the Exchange) is normally or abnormally terminated (see 24.6.1).

3.1.112. open fibre control (OFC): A safety interlock system that controls the optical power level on an open optical fibre cable.

3.1.113. Operation: A construct which may be used by a level above FC-2 and is associated with one or more Exchanges.

3.1.114. Operational: The state of a receiver or transmitter that is capable of receiving or transmitting an encoded bit stream, respectively, based on the rules defined by FC-PH for error control. Those receivers capable of accepting signals from transmitters requiring laser safety procedures are not considered Operational after power on until a signal of a duration longer than that associated with laser safety procedures is present at the fibre attached to the receiver.

3.1.115. Operation_Associator: A value used in the Association_Header to identify a specific operation within a Node and correlate communicating processes related to that operation. Operation_Associator is the mechanism by which an operation within a given Node is referred to by another communicating Node. Operation_Associator is a generic reference to Originator Operation_Associator and Responder Operation_Associator (see Process_Associator and 19.4.2).

3.1.116. optical fall time: The time interval required for the falling edge of an optical pulse to transition between specified percentages of the signal amplitude. For lasers the transitions are measured between the 80% and 20% points. For LED media the specification points are 90% and 10%.

3.1.117. optical fibre: Any filament or fibre, made of dielectric material, that guides light.

3.1.118. optical fiber system test practice (OFSTP): Standards developed and published by the Electronic Industries Association (EIA) under the EIA/TIA-526 series of standards.

3.1.119. optical path penalty: A link penalty to account for those effects other than attenuation.

3.1.120. optical reference plane: The plane that defines the optical boundary between the plug and the receptacle.

3.1.121. optical rise time: The time interval required for the rising edge of an optical pulse to transition between specified perc ntages of the signal amplitude. For lasers the transitions are measured between the 20% and 80% points. For

LED media the specification points ar 10% and 90%.

3.1.122. optical return loss (ORL): The ratio (expressed in units of dB) of optical power incident upon a component port or an assembly to the optical power reflected by that component when that component or assembly is introduced into a link or system.

3.1.123. optional: Characteristics that are not required by FC-PH. However, if any optional characteristic is implemented, it shall be implemented as defined in FC-PH.

3.1.124. ordered set: A Transmission Word composed of a Special Character in its first (leftmost) position and Data Characters in its remaining positions. An Ordered Set is represented by the combination of Special Codes and data bytes which, when encoded, result in the generation of the Transmission Characters specified for the Ordered Set.

3.1.125. Originator: The logical function associated with an N_Port responsible for originating an Exchange.

3.1.126. Originator Exchange Identifier (OX_ID): An identifier assigned by an Originator to identify an Exchange and meaningful only to the Originator (see Responder Exchange Identifier).

3.1.127. Payload: Contents of the Data Field of a frame, excluding Optional Headers and fill bytes, if present (see figure 44, and clauses 17 and 19, and see 20.2).

3.1.128. plug: The cable half of the interface connector which terminates an optical or electrical signal transmission cable.

3.1.129. Port: A generic reference to an N_Port or F_Port.

3.1.130. Port_Name: A Name_Identifier associated with a Port.

3.1.131. power on state: In this state any circuits or optical devices respond to controls resulting from higher levels.

3.1.132. Primitive Sequ nce: An ordered set transmitted repeatedly and continuously until a specified response is received (see 16.4 and 24.8.1).

3.1.133. Primitive Signal: An ordered set designated to have a special meaning such as an Idle or Receiver_Ready (R_RDY) (see 16.3).

3.1.134. Process_Associator: A value used in the Association_Header to identify a process or a group of processes within a Node. Process_Associator is the mechanism by which a process is addressed by another communicating process. Process_Associator is a generic reference to Originator Process_Associator and Responder Process_Associator (see Operation_Associator and clause 19.4.1).

3.1.135. process policy: An error handling policy where an N_Port is able to continue processing Data frames received following detection of one or more missing frames in a Sequence (see 29.6.1.1).

3.1.136. Random Relative Offset: The relationship specified between Relative Offset values contained in frame (n) and frame (n+1) of an Information Category within a single Sequence. For a given Information Category i within a single Sequence, Random Relative Offset (RO_i) value for a frame (n+1) is unrelated to that of the previous frame (n). (see Initial Relative Offset and Continuously Increasing Relative Offset).

3.1.137. receiver: 1. The portion of a Link_Control_Facility dedicated to receiving an encoded bit stream from a fibre, converting this bit stream into Transmission Characters, and Decoding these characters using the rules specified by FC-PH.
2. An electronic circuit (Rx) that converts a signal from the media (optical or electrical) to an electrical retimed (or non-retimed) serial logic signal.

3.1.138. receiver overload: The condition of exceeding the maximum acceptable value of the received average optical power at point R of figure 8 to achieve a BER $< 10^{-12}$.

3.1.139. receiver sensitivity: The minimum acceptable value of average received signal at point R of figure 8 to achieve a BER $< 10^{-12}$. It takes into account power penalties caused by use of a transmitter with a worst-case output. In the cas of an optical path it does not include power penalties associated with dispersion, jitter, effects related to the modal structure of the

source or reflections from the optical path. These effects are specified separately in the allocation of maximum optical path penalty.

3.1.140. receptacle: The fixed or stationary female half of the interface connector which is part of the transmitter or receiver.

3.1.141. reflections: Power returned to point S of figure 8 by discontinuities in the physical link.

3.1.142. RIN_{12}: Laser noise in dB/Hz measured relative to the average optical power with 12dB return loss.

3.1.143. Relative Offset (Offset): The displacement, expressed in bytes, of the first byte of a Payload related to a upper level-defined-origin for a given Information Category (see Continuously Increasing and Random Relative Offset, and 4.14.3).

3.1.144. Relative Offset space: A virtual address space defined by the sending upper level for a single Information Category. The address space starts from zero, representing the upper level-defined-origin, and extends to its highest value.

3.1.145. remote F_Port: The F_Port to which the other communicating N_Port is directly attached (see local F_Port).

3.1.146. repeating ordered set: An Ordered Set which, when issued by FC-2 to FC-1 for transmission, is to be repetitively transmitted until a subsequent transmission request is issued by FC-2.

3.1.147. return loss: See optical return loss.

3.1.148. reserved: A field which is filled with binary zeros by the source N_Port and is ignored by the destination N_Port. Each bit in the reserved field is denoted by "r".

NOTE - Future enhancements to FC-PH may define usages for these reserved fields. The reserved fields should not be checked or interpreted. Any violation of this guideline may result in loss of upward compatibility with future implementations which comply with future enhancements to FC-PH.

3.1.149. Responder: The logical function in an N_Port responsibl for supporting the Exchange initiated by the Originator in another N_Port.

3.1.150. Responder Exchange_Identifi r (RX_ID): An identifier assigned by a Responder to identify an Exchange and meaningful only to the Respond r.

3.1.151. run length: Number of consecutive identical bits in the transmitted signal e.g., the pattern 0011111010 has a run length of five (5).

3.1.152. running disparity: A binary parameter indicating the cumulative Disparity (positive or negative) of all previously issued Transmission Characters.

3.1.153. S_{11}: The ratio of the reflected signal from a port to the incident signal at the port.

3.1.154. Sequence: A set of one or more Data frames with a common Sequence_ID (SEQ_ID), transmitted unidirectionally from one N_Port to another N_Port with a corresponding response, if applicable, transmitted in response to each Data frame (see clause 24).

3.1.155. Sequence_ID (SEQ_ID): An identifier used to identify a Sequence.

3.1.156. Sequence Initiator: The N_Port which initiates a Sequence and transmits Data frames to the destination N_Port (see clause 24).

3.1.157. Sequence Recipient: The N_Port which receives Data frames from the Sequence Initiator and, if applicable, transmits responses (Link_Control frames) to the Sequence Initiator (see clause 24).

3.1.158. Sequence Status Block: A logical construct which tracks the state of a Sequence. Both the Sequence Initiator and the Sequence Recipient have a Sequence Status Block for each concurrently active Sequence (see 4.13.3.2 and 24.8.2).

3.1.159. Solicited Control: One of the Information Categories indicated in the frame header (see 18.2).

3.1.160. Solicited Data: One of the Information Categories indicated in the frame header (see 18.2).

3.1.161. Source_Identifier (S_ID): The address identifier used to indicate the source Port of the transmitted frame.

3.1.162. sourc N_Port: The N_Port from which a frame is transmitted.

3.1.163. Special Charact r: Any Transmission Character considered valid by the Transmission Code but not equated to a Valid Data Byte. Special Characters are provided by the Transmission Code for use in denoting special functions.

3.1.164. Special Code: A code which, when encoded using the rules specified by the Transmission Code, results in a Special Character. Special Codes are typically associated with control signals related to protocol management (e.g., K28.5).

3.1.165. spectral width: 1. FWHM (Full Width Half Maximum) - The absolute difference between the wavelengths at which the spectral radiant intensity is 50 percent of the maximum power. This form is typically used for LED optical sources.
2. RMS - The weighted root mean square width of the optical spectrum. See FOTP-127. This form is typically used for laser optical sources.

3.1.166. streamed Sequence: A new Class 1 or Class 2 Sequence initiated before receiving the final acknowledgement for the previous Sequence in the same Exchange. Any new Class 3 Sequence initiated before the expiration of R_A_TOV for all Data frames in the previous Sequence (see 18.6).

3.1.167. subblock: A upper level construct which contains partial application data for a single Information Category (see block). A collection of subblocks for a given Information Category may be specified for transfer within a single Sequence.

3.1.168. synchronization: Receiver identification of a Transmission-Word boundary.

3.1.169. TNC: Acronym for a Threaded-Neil-Councilman Coaxial Cable Connector. Specifications for TNC style connectors are defined in MIL-C-39012 and MIL-C-23329.

3.1.170. Transmission Character: Any encoded character (valid or invalid) transmitted across a physical interface specified by FC-0. Valid Transmission Characters are specified by the Transmission Code and include Data and Special Characters.

3.1.171. Transmission Code: A means of Encoding data to enhance its Transmission Characteristics. The Transmission Code specified by FC-PH is byte-oriented, with (1) Valid Data Bytes and (2) Special Codes encoded into 10-bit Transmission Characters.

3.1.172. Transmission Word: A string of four contiguous Transmission Characters occurring on boundaries that are zero modulo 4 from a previously received or transmitted Special Character.

3.1.173. transmitter: 1. The portion of a Link_Control_Facility dedicated to converting Valid Data Bytes and Special Codes into Transmission Characters using the rules specified by the Transmission Code, converting these Transmission Characters into a bit stream, and transmitting this bit stream onto the transmission medium (optical or electrical).
2. An electronic circuit (Tx) that converts an electrical logic signal to a signal suitable for the communications media (optical or electrical).

3.1.174. transceiver: A transmitter and receiver combined in one package.

3.1.175. Uncategorized Information Category: One of the Information Categories indicated in the frame header (see 18.2).

3.1.176. unrecognized ordered set: A Transmission Word containing a K28.5 in its first (leftmost) position but not defined to have meaning by FC-PH.

3.1.177. Unsolicited Control: One of the Information Categories indicated in the frame header (see 18.2).

3.1.178. Unsolicited Data: One of the Information Categories indicated in the frame header (see 18.2).

3.1.179. upper level: A level above FC-2.

3.1.180. Upper Level Protocol (ULP): The protocol user of FC-4 (see clause 4).

3.1.181. valid: A validity control bit indicates if a field is valid, in which case, the value in the field shall be treated as valid. If a validity control bit indicates that a field is invalid, the value in the field shall be treated as invalid (see meaningful).

3.1.182. Valid Data Byte: A string of eight contiguous bits within FC-1 which represents a value with 0 to 255, inclusive.

3.1.183. Valid frame: A frame received with a valid Start_of_Frame (SOF), a valid End_of_Frame (EOF), valid Data Characters, and proper Cyclic Redundancy Check (CRC) of the Frame Header and Data Field (see 17).

3.1.184. vendor unique: Functions, code values, and bits not defined by FC-PH and set aside for private usage between parties using FC-PH. Caution: Different implementations of FC-PH may assign different meanings to these functions, code values, and bits.

3.1.185. well-known addresses: A set of address identifiers defined in FC-PH to access global server functions such as a name server (see 18.3).

3.1.186. word: A string of four contiguous bytes occurring on boundaries that are zero modulo 4 from a specified reference.

3.1.187. Worldwide_Name: An Name_Identifier which is worldwide unique, and represented by a 64 bit unsigned binary value.

3.2 Editorial conventions

In this standard, a number of conditions, mechanisms, sequences, parameters, events, states, or similar terms are printed with the first letter of each word in upper-case and the rest lower-case (e.g., Exchange, Class). Any lower case uses of these words have the normal technical English meanings.

Numbered items in this standard do not represent any priority. Any priority is explicitly indicated.

In case of any conflict between figure, table, and text, the text takes precedence. Exceptions to this convention are indicated in the appropriate sections.

In all of the figures, tables, and text of this document, the most significant bit of a binary quantity is shown on the left side. Exceptions to this convention are indicated in the appropriate sections.

The term "shall" is used to indicate a mandatory rule. If such a rule is not followed, the results are unpredictable unless indicated oth rwise.

The fields or control bits which are not applicable shall be reset to zero.

If a field or a control bit in a frame is specified as not meaningful, the entity which receives the frame shall not check that field or control bit.

Hexadecimal notation
Hexadecimal notation is used to represent fields. For example, a three byte D_ID field containing a binary value of 11111111 11111111 11111010 is denoted by FF FF FA.

3.3 Abbreviations, acronyms, and symbols

Abbreviations, acronyms and symbols applicable to this standard are listed. Definitions of several of these items are included in 3.1. The index at the back of the document is an aid to help locate these terms in the body of the document.

3.3.1 Data rate abbreviations

Abbreviation	Data Rate
133 Mb/s	132,812 5 Mb/s
266 Mb/s	265,625 Mb/s
531 Mb/s	531,25 Mb/s
1 063 Mb/s	1 062,5 Mb/s

The exact data rates are used in the tables and the abbreviated form is used in text. Note that 1,062 5 Gbaud is the preferred ISO method and is used instead of 1 062,5 Mbaud where it makes sense to do so.

3.3.2 Acronyms and other abbreviations

ABTS	Abort Sequence
ABTX	Abort Exchange
ACC	Accept
ACK	Acknowledgement
ADVC	Advise Credit
alias	alias address identifier
BB_Credit	Buffer-to-buffer Credit
BB_Credit_CNT	Buffer-to-buffer Credit_Count

BER	Bit Error Rate
BNC	Bayonet-Neil-Councilman (coaxial connector)
BSY	busy
CATV	Central Antenna Television
CCITT	Comite Consultatif International Telegraphique et Telephonique (see ITU-TS)
CD	compact disk
Class 1/SOF$_{c1}$	Class 1 frame with a **SOF$_{c1}$** delimiter
Credit_CNT	Credit_Count
dB	decibel
dBm	decibel (relative to 1 mw power)
DF_CTL	Data_Field Control
D_ID	Destination_Identifier
DJ	deterministic jitter
DUT	device under test
ECL	Emitter Coupled Logic
E_D_TOV	Error_Detect_Timeout value
EE_Credit	End-to-end Credit
EE_Credit_CNT	End-to-end Credit_Count
EIA	Electronic Industries Association
EMC	electromagnetic compatibility
EOF	End-of-Frame
ESB	Exchange Status Block
ESTC	Estimate Credit
ESTS	Establish Streaming
F_BSY	Fabric_Port_Busy
F_BSY(DF)	F_BSY response to a Data frame
F_BSY(LC)	F_BSY response to any Link_Control except P_BSY
FC	Fibre Channel
FCS	Frame Check Sequence (see N.1)
F_CTL	Frame Control
FOTP	fiber optic test procedure
F_Port	Fabric_Port
F_RJT	Fabric_Port_Reject
FT-0	frame type 0
FT-1	frame type 1
FWHM	Full Width Half Max
hex	hexadecimal notation
Hz	Hertz = 1 cycle per second
Idle	Idle Word
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ITU-TS	The International Union - Telecommunication Standartization (formerly CCITT)
LA_RJT	Link Application Reject
LCF	Link_Control_Facility
LCR	Link Credit Reset
LED	light emitting diode
LOGI	Login
LOGO	Logout
LOL	loss of light
LR	Link Reset Primitive Sequence
LRR	Link Reset Response Primitive Sequence
LW	long wavelength
m	meter
MAS	master of link
Mb	Mega bit
MB	Mega Byte
MBd	Mega Baud
ms	millisecond
μs	microsecond
MM	multimode
NA	not applicable
NAA	Network_Address_Authority
NOP	No Operation
NOS	Not_Operational Primitive Sequence
N_Port	Node_Port
ns	nanosecond
OESB	Originator Exchange Status Block
OFC	open fibre control
Offset	Relative Offset
OFSTP	optical fiber system test practice
OLS	Offline Primitive Sequence
ORL	optical return loss
OX_ID	Originator Exchange_Identifier
P_BSY	N_Port_Busy
ppm	parts per million
P_RJT	N_Port_Reject
R_A_TOV	Resource_Allocation_Timeout value
RCS	Read Connection Status
R_CTL	Routing Control
RES	Read Exchange Status Block
RESB	Responder Exchange Status Block
RFI	Radio Frequency Interference
RIN	relative intensity noise
RIIN	reflection induced intensity noise
RJ	random jitter
RJT	reject
RMC	Remove Connection
RMS	root mean square
RO	Relative Offset
R_RDY	Receiver_Ready
RSS	Read Sequence Status Block
R_T_TOV	Receiver_Transmitter_Timeout value
RTV	Read Timeout Value
Rx	receiver
RX_ID	Responder Exchange_Identifier
SBCCS	Single Byte Command Code Sets
s.	second
SEQ_CNT	Sequence_Count
SEQ_ID	Sequence_ID
S_ID	Source_Identifier
S/N	signal-to-nois ratio
SISB	Sequence Initiator Status Block

S_Length	Security_Length
SM	single mode
SOF	Start-of-Frame
SRSB	Sequence Recipient Status Block
SSB	Sequence Status Block
S_Type	Security_Type
STP	shielded twisted pair
SW	short wavelength
TNC	Threaded-Neil-Councilman (coaxial connector)
TP	twisted pair
Tx	transmitter
TYPE	Data structure type
UI	unit interval = 1 bit period
ULP	Upper Level Protocol
X_ID	Exchange_Identifier
WWN	Worldwide_Name

3.3.3 Symbols

Unless indicated otherwise, the following symbols have the listed meaning.

@	at
||	concatenation
Ω	ohms
μ	micro (e.g., μm = micrometre)

4 Structure and concepts

This clause provides an overview of the structure, concepts and mechanisms used in FC-PH and is intended for informational purposes only.

The Fibre Channel (FC) is logically a bidirectional point-to-point serial data channel, structured for high performance capability. Physically, the Fibre Channel can be an interconnection of multiple communication points, called N_Ports, interconnected by a switching network, called a Fabric, or a point-to-point link. Fibre is a general term used to cover all physical media types supported by the Fibre Channel - such as optical fibre, twisted pair, and coaxial cable.

Fibre Channel is structured as a set of hierarchical functions as illustrated in figure 2. Fibre Channel Physical and Signalling interface (FC-PH) consists of related functions FC-0, FC-1, and FC-2. Each of these functions is described as a level. FC-PH does not restrict implementations to specific interfaces between these levels.

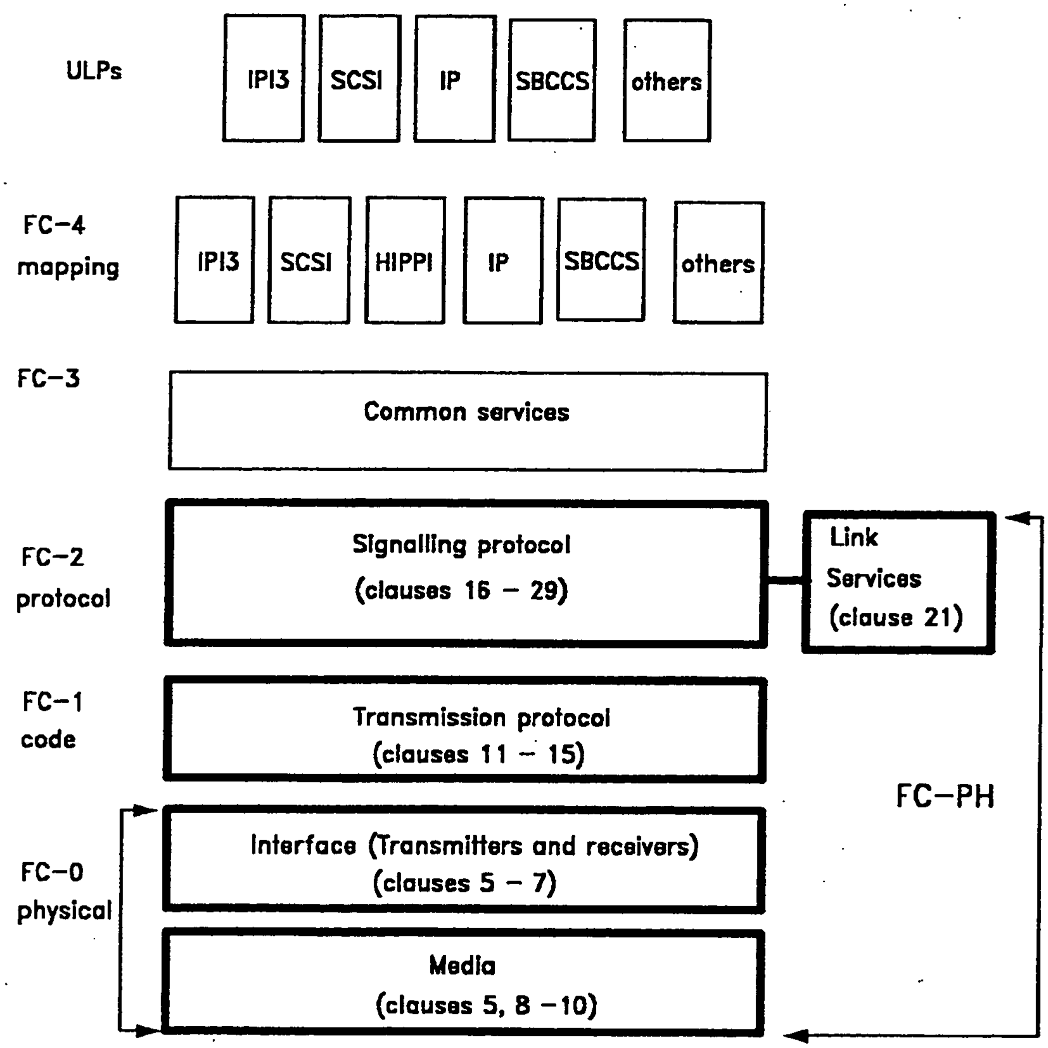


Figure 2 - Fibre Channel structure

The Physical interface (FC-0) consists of transmission media, transmitters, and receivers and their interfaces. The Physical interface specifies a variety of media, and associated drivers and receivers capable of operating at various speeds. The transmission code (FC-1) used is 8B/10B and is specified in clause 11.

The signalling protocol (FC-2) specifies the rules, and provides mechanisms needed to transfer blocks of data end to end. FC-2 defines a suite of functions and facilities available for use by an FC-4. This suite of functions and facilities may be more than what a given FC-4 may require. Based on their needs, an FC-4 may choose only a subset of these functions and facilities.

FC-3 provides a set of services which are common across multiple N_Ports of an FC node. An Upper Level Protocol mapping to FC-PH constitutes an FC-4 which is the highest level in the Fibre Channel structure.

Fibre Channel provides a method for supporting a number of Upper Level Protocols (ULPs). The Link Services represent a mandatory function required by FC-2 (see clause 21).

A Fibre Channel Node is functionally configured as illustrated in figure 3. A Node may support one or more N_Ports and one or more FC-4s. Each N_Port contains FC-0, FC-1 and FC-2 functions. FC-3 optionally provides the common services to multiple N_Ports and FC-4s.

4.1 FC-0 general description

The FC-0 level of FC-PH describes the Fibre Channel link. The FC-0 level covers a variety of media and the associated drivers and receivers capable of operating at a wide range of speeds. The FC-0 level is designed for maximum flexibility and allows the use of a large number of technologies to meet the widest range of system requirements.

Each fibre is attached to a transmitter of a Port at one end and a receiver of another Port at the other end (see figure 4). When a Fabric is present in the configuration, a fibre may attach to an N_Port and an F_Port (see figure 7). Patch panels or portions of the active Fabric may function as repeaters, concentrators or fibre converters.

A communicating route between two N_Ports may be made up of links of different technologies. For example, it may have multimode fibre links attached to end Ports but may have a single mode link in between as illustrated in figure 5. In figure 6, a typical Fibre Channel building wiring configuration is shown.

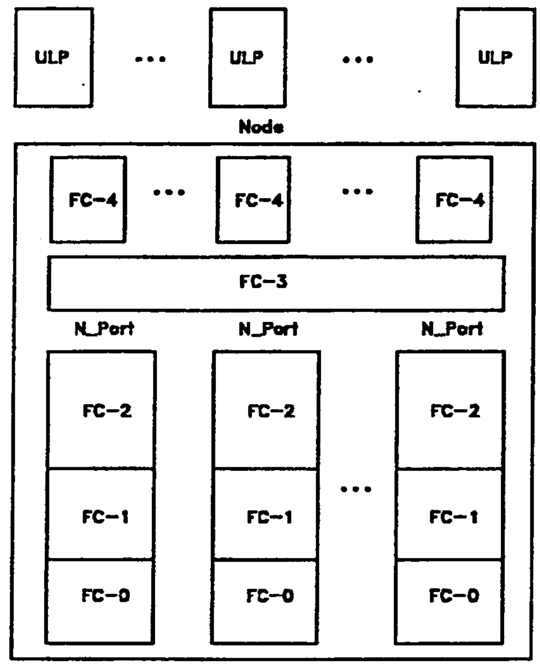


Figure 3 - Node functional configuration

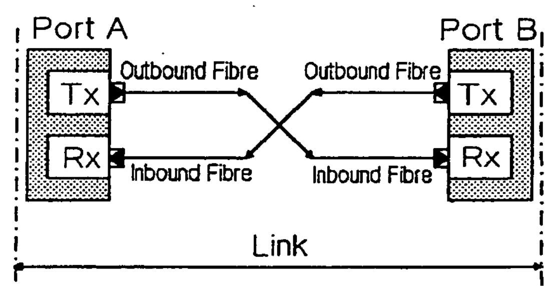


Figure 4 - FC-0 link

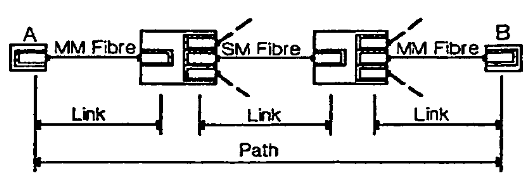


Figure 5 - FC0 path

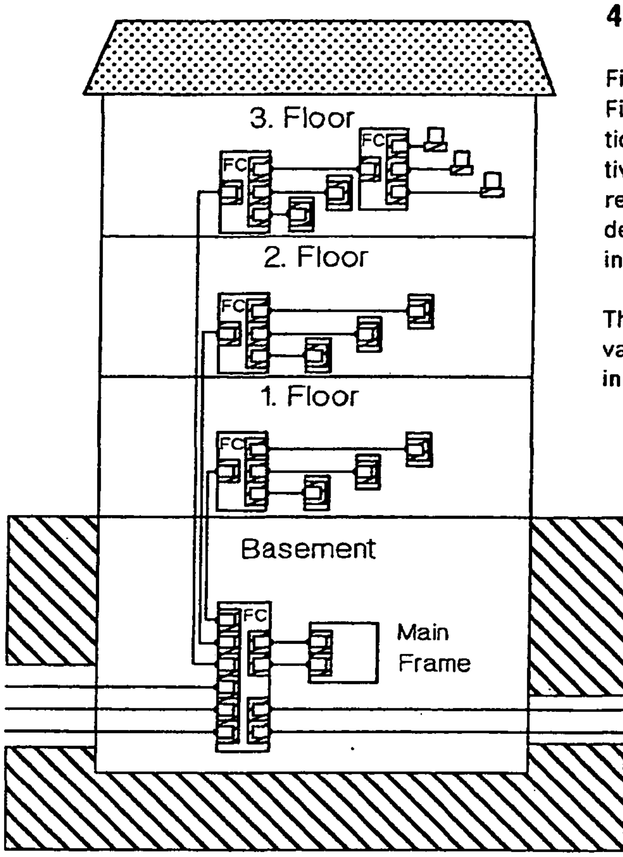


Figure 6 - Fibre Channel building wiring

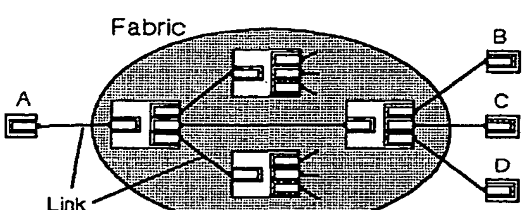


Figure 7 - Fabric

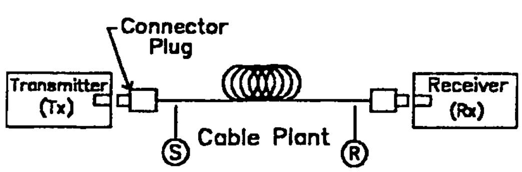


Figure 8 - FC-0 cable plant

4.2 FC-0 int rfac overview

Figure 8 shows the cable plant configuration. Figures 9 and 10 illustrate a set of implementations of the transmitter and receiver, respectively. Interfaces "a" through "e" are for reference only and are implementation dependent. Recommended interfaces are included in annex D.

The nomenclature used by FC-0 to reference various combinations of components is defined in clause 5.

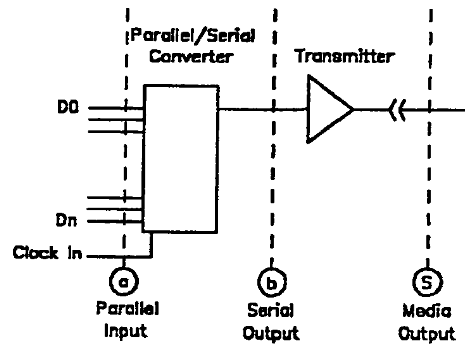


Figure 9 - FC-0 transmitter interfaces

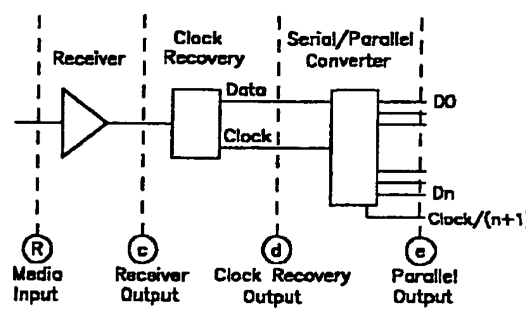
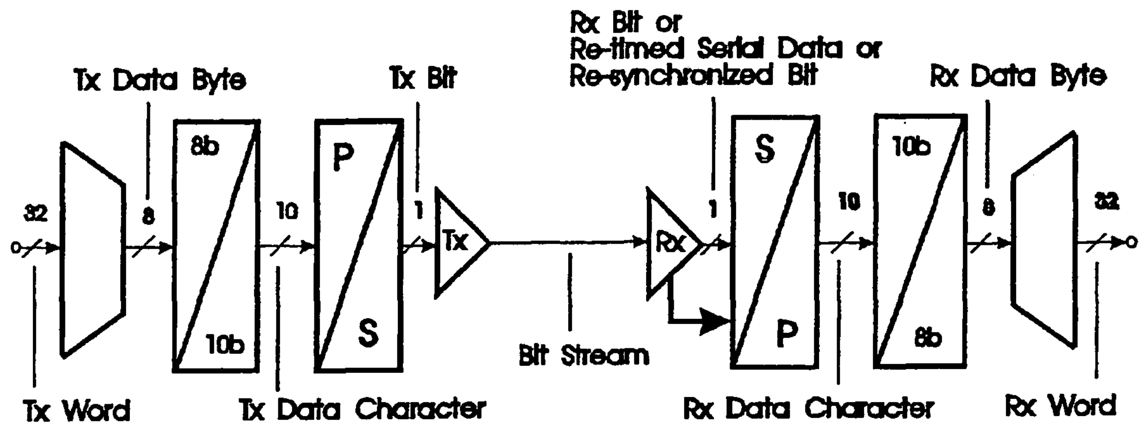


Figure 10 - FC-0 receiver interfaces

The link distance capability specified in FC-PH is based on ensuring interoperability across multiple vendors supplying the technologies (both link transceivers and cable plants) under the tolerance limits specified in FC-PH. Greater link distances may be obtained by specifically engineering a link based on knowledge of the technology charact ristics and the conditions under which the link is installed and operated. However, such link distance extensions are outside the scope of FC-PH.



Legend:

Tx - transmitter	Rx - receiver
Tx Word - 32 bit transmit word	Rx Word - 32 bit receive word
Tx Byte - 8 bit transmit byte	Rx Byte - 8 bit receive byte
Tx Bit - 1 transmit bit	Rx Bit - 1 receive bit

P - parallel side S - serial side

Figure 11 - Data flow stages

The user needs to take care that their use conditions at least conform to the specified conditions of the document.

Data flow stages
Figure 11 illustrates the data flow stages of 32 bit word parallel, 8 bit byte parallel, 10 bit character parallel, and bit serial and vice versa.

4.3 FC-1 general description

The Fibre Channel transmits information using an adaptive 8B/10B code to bound the maximum run length of the code, maintain DC-balance, and provide word alignment. The encoding process results in the generation of Transmission Characters. Two types of Transmission Characters (Data and Special) are defined.

Certain combinations of Transmission Characters, referred to as Ordered Sets, are designated by FC-PH to have special meaning. Ordered Sets are used to identify frame boundaries, transmit primitive function requests, and maintain proper link transmission characteristics during periods of inactivity.

Transmitter and Receiver behavior is specified via a set of states and their interrelationships. These stat s ar divided into Operational and Not Operational classes. Error monitoring capabilities and special operational modes are also defined for Operational Receivers and Transmitters.

4.4 FC-2 general description

The FC-2 level serves as the transport mechanism of the Fibre Channel. The transported data is transparent to FC-2 and visible to FC-3 and above.

The following concepts are defined and described:

- Node and N_Port and their identifiers
- communication models.
- Topologies based on the presence or absence of a Fabric.
- Classes of service provided by the Fabric and the N_Ports.
- General Fabric model
- Building Blocks and their hierarchy
- Sequence and Sequence Identifier
- Exchange and Exchange Identifier
- Sequence Status Block model
- Exchange Status Block model
- Segmentation and reassembly

A Request-Reply protocol example is used in several clauses for illustration but other protocols are equally valid.

4.5 FC-PH physical model

Figure 12 depicts the FC-PH physical model and illustrates the FC-PH physical structure and components. The Fibre Channel (FC) physically consists of a minimum of two Nodes, each with a minimum of one N_Port interconnected by a pair of fibres - one outbound and the other inbound at each N_Port. This pair of unidirectional fibres transmitting in opposite directions with their associated transmitters and receivers is referred to in FC-PH as a link. The link is used by the interconnected N_Ports to perform data communication.

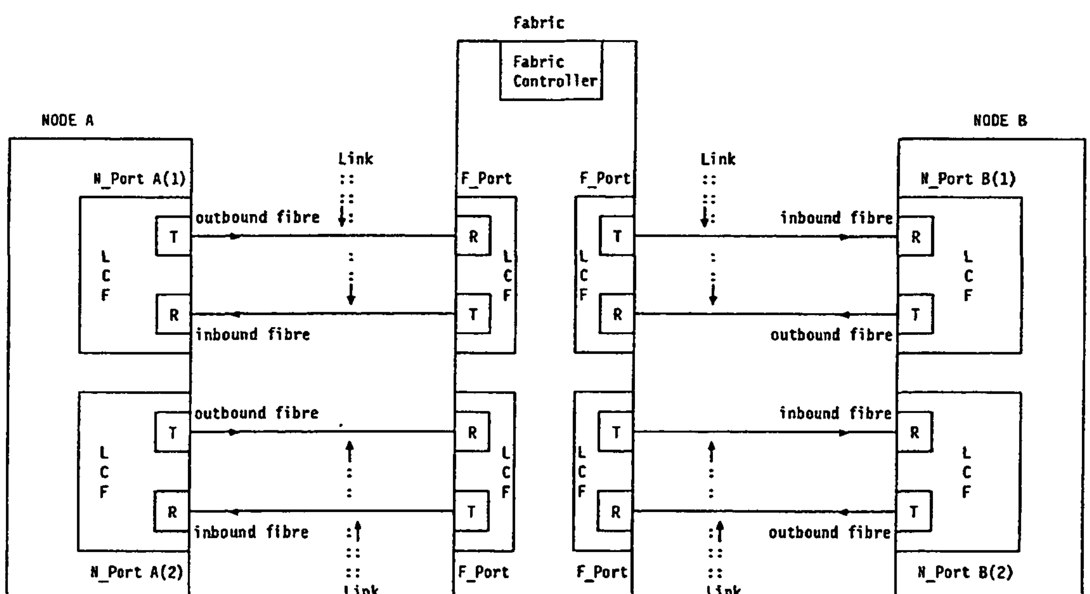
Physical equipment such as a processor, controller, or terminal can be interconnected to other physical equipment through these links. Attached physical equipment supports one or more Nodes and each Node contains one or more N_Ports, with each N_Port containing a transmitter and a receiver.

The physical model shown in figure 12 is inherently capable of simultaneous, symmetrical bidirectional flow. A Fabric may be present between the N_Ports and some Fabrics may not support this type of flow. From the perspective of a given N_Port, for instance A(1) or B(1), its transmitter sends Data frames on the outbound fibre and its receiver receives the responses on the inbound fibre.

In Class 1 service, an N_Port logically performs a point-to-point communication with another N_Port at any given time. This statement is true regardless of the presence of Fabric. However, multiple N_Ports in a Node can simultaneously perform data transfers in parallel with single or multiple N_Ports contained in one or more Nodes (see 4.9.1 and clause 22 for Class of service description).

In Class 2 and Class 3 service, an N_Port may multiplex frames to or demultiplex frames from multiple N_Ports (see 4.9.2, 4.9.3, and clause 22 for Class of service description).

This structure provides the attached equipment flexible mechanisms to perform simultaneous data transfers in parallel, to or from, single or multiple equipments.



Legend:
T = Transmitter
R = Receiver
Fibre = Any medium type supported by Fibre Channel
LCF = Link_Control_Facility
A(1), A(2) = N_Port addresses within Node A
B(1), B(2) = N_Port addresses within Node B

Figure 12 - FC-PH physical model

4.5.1 Node and N_P rt identifiers

Each N_Port has an identifier to identify itself during communication. Referring to figure 12, Node A consists of two N_Ports represented as A(1) and A(2) and Node B consists of two N_Ports represented as B(1) and B(2). Thus, single or multiple N_Ports contained within a Node are represented by a set of N_Port Identifiers. Each N_Port is represented as an element of the set, for example,

Node B as {B(1), B(2), ..., B(n)}.

4.5.2 Link_Control_Facility (LCF)

The Link_Control_Facility is a hardware facility which attaches to each end of a link and manages transmission and reception of data. It is contained within each N_Port or F_Port and includes a transmitter and a receiver. Each LCF provides the logical interface to the Originator and/or the Responder, as applicable to a communication model.

4.6 Communication models

An N_Port transmits Data frames as a result of information transfer requests from its associated FC-4s at its end, and transmits Link_Control responses to Data frames that it receives from other N_Ports. An N_Port receives Link_Control responses to Data frames that it transmitted, and it receives Data frames from other N_Ports.

An N_Port may operate according to the communication models described below (see annex L).

- Model 1 operation is defined as an N_Port transferring Data frames in one direction only, with Link_Control frames flowing in the opposite direction.
- Model 2 operation is defined as an N_Port simultaneously transmitting and receiving Data frames, with Link_Control frames flowing in both directions as well.
- Model 3 operation is defined as an N_Port both transmitting and receiving data, but not simultaneously. Data frames and Link_Control frames flow in both directions, but the flow is limited, possibly by a Fabric, to a single direction at a time.

4.7 Bandwidth

Effective transfer rate is a function of physical variants such as the communication model, Payload size, Fibre speed, and overhead specified by FC-PH. Examples of bandwidth for a selected set of variants are shown in table 1 (see annex M).

Table 1 - Bandwidth examples

Communication model	Speed	Payload size (bytes)	Bandwidth (MB/sec)
1	1,062 5 Gbaud	2 048	103,22 1)
2	1,062 5 Gbaud	2 048	100,37 2)
3	1,062 5 Gbaud	2 048	100,37 3)

Note:
1) Aggregate for both Fibres
2) Per direction
3) Aggregate for all communicating N_Ports

4.8 Topology

Topologies are defined, based on the capability and the presence or absence of Fabric between the N_Ports, and they are:

- Point-to-point topology
- Fabric topology
- Arbitrated Loop topology

FC-PH protocols are topology independent. Attributes of a Fabric may restrict operation to certain communication models.

4.8.1 Point-to-point topology

The topology shown in figure 13, in which communication between N_Ports occurs without the use of Fabric, is defined as point-to-point.

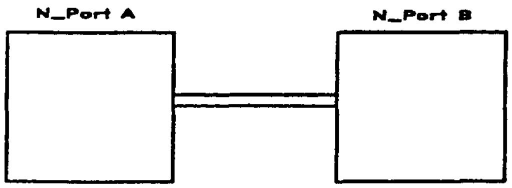


Figure 13 - Point-to-point topology

4.8.2 Fabric topology

This topology uses the Destination_Identifier (D_ID) embedded in the Frame_Header to route the frame through a Fabric to the desired destination N_Port. Figure 14 illustrates multiple N_Ports interconnected by a Fabric.

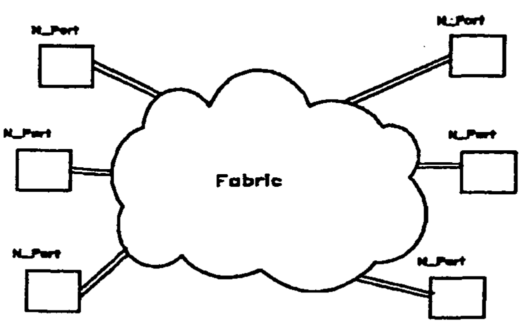


Figure 14 - Fabric topology

4.8.3 Arbitrated Loop topology

The Arbitrated Loop topology permits three or more L_Ports to communicate without the use of a Fabric, as in Fabric topology. The arbitrated loop supports a maximum of one point-to-point circuit at a time. When two L_Ports are communicating, the arbitrated loop topology supports simultaneous, symmetrical bidirectional flow.

Figure 15 illustrates two independent Arbitrated Loop configurations each with multiple L_Ports attached. Each line in the figure between L_Ports represents a single fibre. The first configuration shows an Arbitrated Loop composed only of NL_Ports. The second configuration shows an Arbitrated Loop composed of one FL_Port and three NL_Ports.

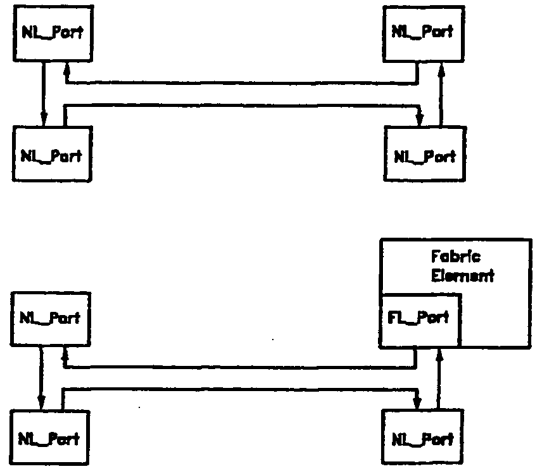


Figure 15 - Examples of the Arbitrated Loop Topology

4.9 Classes of service

Three Classes of service are defined. These Classes of service are distinguished primarily by the methodology with which the communication circuit is allocated and retained between the communicating N_Ports and the level of delivery integrity required for an application. Classes of service are topology independent. If the Fabric is not present, the service is provided as a special case of point-to-point. Fabrics and N_Ports are not required to support all Classes of service.

4.9.1 Class 1 service - Dedicated Connection

Class 1 is a service which establishes Dedicated Connections. Once established, a Dedicated Connection is retained and guaranteed by the Fabric. This service guarantees maximum bandwidth available between two N_Ports across the established connection.

In Class 1, frames are delivered to the destination N_Port by the Fabric in the same order as they are transmitted by the source N_Port. If the Fabric is not present, this service becomes a special case of point-to-point.

4.9.2 Class 2 service - Multiplex

Class 2 is a Connectionless service with the Fabric multiplexing frames at frame boundaries. If a Fabric is not present, this service becomes a special case of point-to-point.

The transmitter transmits Class 2 Data frames in a sequential order within a given Sequence. However the Fabric may not guarantee the order of delivery and frames may be delivered out of order.

The Fabric guarantees notification of delivery or failure to deliver in the absence of link errors. In case of link errors, notification is not guaranteed since the Source_Identifier (S_ID) may not be error free.

4.9.3 Class 3 service - Datagram

Class 3 is a Connectionless service with the Fabric multiplexing frames at frame boundaries, if a Fabric is present. If a Fabric is not present, this service becomes a special case of point-to-point.

Class 3 supports only unacknowledged delivery where the destination N_Port does not send any confirmation Link_Control frames on receipt of valid Data frames. Any acknowledgement of Class 3 service is up to and determined by ULPs.

The transmitter transmits Class 3 Data frames in sequential order within a given Sequence. However, the Fabric may not guarantee the order of delivery and frames may be delivered out of order.

In Class 3, the Fabric is expected to make a best effort to deliver the frame to the intended destination and does not issue a busy or reject frame to the source N_Port if unable to deliver the frame.

When a Class 3 frame is received in error, any error recovery or notification is performed at the ULP level.

4.10 Intermix

Intermix is an option of Class 1 service which allows interleaving of Class 2 and Class 3 frames during an established Class 1 Connection between two N_Ports. During a Class 1 Connection, an N_Port capable of Intermix may transmit and receive Class 2 and Class 3 frames interleaved with Class 1 frames. Class 2 and Class 3 frames may be interchanged between the two connected N_Ports or between either of the connected N_Ports and other N_Ports. Support for the Intermix option of Class 1 service is indicated during Login (see clause 23). An N_Port which supports Intermix is capable of both transmitting and receiving intermixed frames.

In a point-to-point topology, both interconnected N_Ports are required to support Intermix if Intermix is to be used. In the presence of a Fabric, both the N_Port and the F_Port to which it is attached are required to support Intermix if Intermix is to be used. Fabric support for Intermix requires that the full Class 1 bandwidth during a Dedicated Connection be maintained. Intermix permits the potentially unused Class 1 bandwidth to be used for transmission of Class 2 and Class 3 frames.

4.11 General Fabric model

The primary function of the Fabric is to receive the frames from a source N_Port and route the frames to the destination N_Port whose address identifier is specified in the frames. Each N_Port is physically attached through a link to the Fabric. FC-2 specifies the protocol between the Fabric and the attached N_Ports. A Fabric is characterized by a single address space in which every N_Port has a unique N_Port identifier.

A Fabric specifies the Classes of service it supports in its Service Parameters (see 23.7). Fabrics are allowed to provide the Classes of service through equivalent mechanisms and/or functions. Refer to the Fabric requirements (FC-FG) document for these equivalent functions provided by some Fabrics.

Figure 16 illustrates the general Fabric model. The model is conceptual and may provide the following major functions:

- Bidirectional Fabric Ports (F_Ports)

- Receive buffer
- Connection Sub-Fabric
- Connectionless Sub-Fabric
- Receive buffer queue management

4.11.1 Fabric Ports (F_Ports)

The Fabric model contains two or more F_Ports. Each F_Port can be attached to an N_Port through a link. Each F_Port is bidirectional and supports one or more communication models.

The receiving F_Port responds to the sending N_Port according to the FC-2 protocol. The

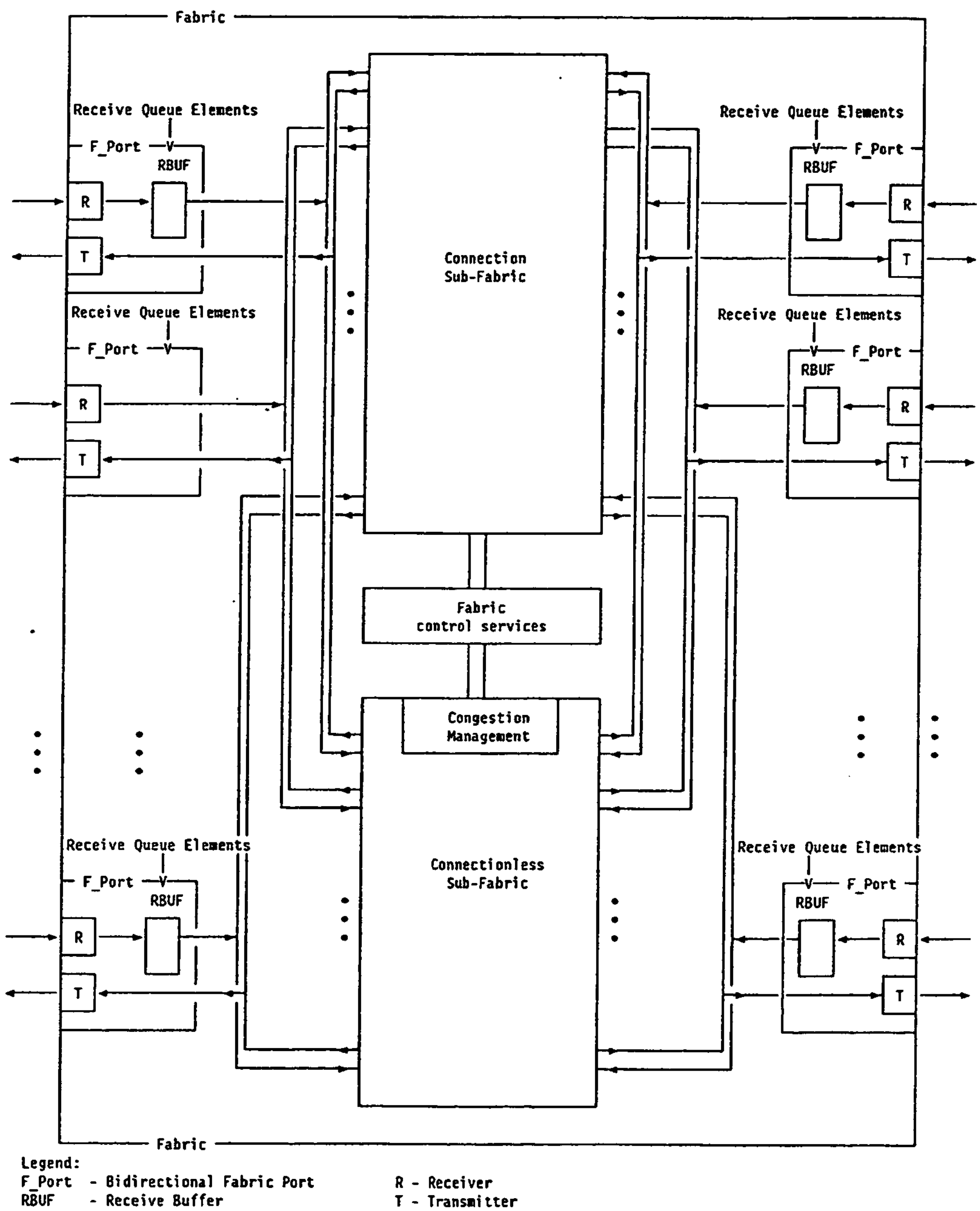


Figure 16 - Informative g n ral Fabric model

Fabric may or may not verify the validity of the frame as it passes through the Fabric (see 17.6.2 and 17.8.1).

An F_Port may or may not contain receive buffers for the incoming frames. The maximum frame size that the Fabric can handle for Class 2 service, Class 3 service, and for the frame which requests Class 1 service is determined during Login. One of the Fabric Service Parameters indicates the maximum frame size for the entire Fabric.

The Fabric routes the frame to the F_Port directly attached to the destination N_Port based on the N_Port identifier (D_ID) embedded in the frame. The address translation and the routing mechanisms within the Fabric are transparent to N_Ports and are not specified in FC-PH.

4.11.2 Connection based Sub-Fabric

The Connection based Sub-Fabric provides Dedicated Connections between F_Ports and the N_Ports attached to these F_Ports. Such Dedicated Connections are bidirectional and support full transmission rate concurrently in both directions. A Dedicated Connection is retained until a removal request is received from one of the communicating N_Ports or an exception condition occurs which causes the Fabric to remove the Connection.

On receiving a request from an N_Port, the Fabric establishes a Dedicated Connection to the destination N_Port through the Connection Sub-Fabric. The mechanisms used by the Fabric to establish the connection are transparent to FC-2.

The Sub-Fabric is not involved in flow control, which is managed end-to-end by the N_Ports.

If the Fabric is unable to establish a Dedicated Connection, it returns a busy or reject frame with a reason code. Once the Dedicated Connection is established, the frames between the communicating N_Ports are routed through the same circuit and all responses are issued by N_Ports.

A Class 1 N_Port and the Fabric may support stacked connect-requests. This stacked connect-requests support allows an N_Port to request multiple Dedicated Connections to multiple destinations and allows the Fabric to service them in any order. While the N_Port is connected to one destination, another connect-request may be processed by the Fabric to minimize the connect latency.

Unless a Class 1 N_Port and the Fabric support stacked connect-requests, when two N_Ports are in Dedicated Connection, a Class 1 connect-request from another source for one of these N_Ports is busied regardless of Intermix support.

If a Class 2 frame destined to one of the N_Ports established in a Dedicated Connection is received, the Class 2 frame may be busied and the transmitting N_Port is notified. In the case of a Class 3 frame, the frame is discarded and no notification is sent. The destination F_Port may be able to hold the frame for a period of time before discarding the frame or returning a busy frame. If Intermix is supported and the Fabric receives a Class 2 or Class 3 frame destined to one of the N_Ports established in a Dedicated Connection, the Fabric may allow delivery with or without a delay if the delivery does not interfere with the transmission of Class 1 frames (see 22.4).

4.11.3 Connectionless Sub-Fabric

A Connectionless Sub-Fabric is characterized by the absence of Dedicated Connections. The Connectionless Sub-Fabric multiplexes frames at frame boundaries between an F_Port and any other F_Port and between the N_Ports attached to them.

The Fabric notifies the transmitting N_Port with a reason code embedded in a Link_Response frame, if it is unable to deliver a Class 2 frame. In the case of a Class 3 frame, the Fabric does not notify the transmitting N_Port.

A given frame flows through the Connectionless Sub-Fabric for the duration of the routing. After the frame is routed, the Connectionless Sub-Fabric is not required to have memory of source, routing, or destination of the frame.

When frames from multiple N_Ports are targeted for the same destination N_Port in Class 2 or Class 3, cong stion of frames may occur within the Fabric. Management of this congestion is part of the Connectionless Sub-Fabric and buffer-to-buffer flow control.

If any buffer-to-buffer flow control error occurs and as a result causes overflow (see clause 26), the Fabric logs the error and may discard the overflow frame without notification. Error logging is vendor unique.

4.12 Fibre Channel services

Fibre Channel services (e.g., Directory Server) may be provided in a Fibre Channel configuration to meet the needs of the configuration. Each of these services is addressed with a well-known address for the N_Port providing the service (see 18.3). These well-known addresses are recognized and routed to by the Fabric. These services may be centralized or distributed.

4.13 Building Blocks

The set of building blocks defined in FC-2 are:

- Frame
- Sequence
- Exchange
- Protocol

4.13.1 Building block hierarchy

The FC-2 building blocks are used in a hierarchical fashion, as illustrated in figure 17.

A Sequence is made up of one or more Data frames and if applicable, corresponding responses (see 24.1 and clause 20). An Exchange is made up of one or more Sequences flowing in a single direction from the Originator of the Exchange to the Responder or in both directions between the Originator and the Responder (see clause 24).

Prior to use by a ULP for its data transfer, the Fibre Channel has to be setup for the operating environment. The Fibre Channel operating environment is setup by performing two protocols named Fabric Login and N_Port Login (see clause 23). Once these two Logins are performed, a ULP may start using the Fibre Channel until one or both of these Logins are invalidated. For example, after power down of an N_Port, the previous Fabric Login might have become invalid due to an event such as Fabric reconfiguration and the N_Port may need to repeat the Fabric Login.

Each Login uses an Exchange as the mechanism to accomplish th function. A ULP data transfer is performed using an Exchange as the mechanism (see figure 17) with the related FC-4 translating the ULP protocol to FC-2 protocol.

Protocol examples
Some examples of data transfer protocol and N_Port Login are included in annex P.

4.13.2 Frame

Frames are based on a common frame format (see clause 17). Frames are broadly categorized as Data frames and Link_Control frames (see table 47). Data frames (see 20.2) are classified as

- Link_Data frames,
- Device_Data frames, and
- Video_Data frames.

Link_Control frames (see 20.3) are classified as

- Acknowledge (ACK) frames,
- Link_Response (Busy and Reject) frames, and
- Link_Command frames.

Selective retransmission of frames for error recovery is not supported in FC-PH (see clause 29). However, an individual frame may be busied in Class 2 and the sender retransmits the busied frame (see 20.3.3.1 and 20.3.3.2) up to the ability of the sender to retry (see clause 22). The number of times the sender may retry is not specified in FC-PH and may be application or vendor unique.

4.13.3 Sequence

A Sequence is a set of one or more related Data frames transmitted unidirectionally from one N_Port to another N_Port with corresponding Link_Control frames, if applicable, transmitted in response. An N_Port which transmits a Sequence is referred to as the Sequence Initiator and the N_Port which receives the Sequence is referred to as the Sequence Recipient. Sequence rules are specified in clause 24.

Error recovery is performed on the Sequence boundary at the discretion of a level higher than FC-2. If a frame is not transmitted error free, and the rror policy requires error recovery, the

Sequ nce to which the frame belongs may be retransmitted (see clause 29).

4.13.3.1 Sequence_Identifier (SEQ_ID)

The Sequence Initiator assigns the Sequence a Sequence_Identifier (SEQ_ID). The Sequence Recipient uses the same SEQ_ID in its response frames. The Sequence Initiator at each of the communicating N_Port pair assigns SEQ_ID independent of the other.

4.13.3.2 Sequence Status Blocks

A Sequence Status Block (SSB) is a logical construct representing the format of the Sequence status information (see 24.8). It is used to track the progress of a Sequence at an N_Port on a frame by frame basis. A Sequence Initiator SSB and a Sequence Recipient SSB are used by the respective N_Ports to track the status of a given Sequence.

- When a Sequence Initiator starts a Sequence, the Sequence Initiator allocates a SSB to be associated with the Sequence it has initiated.

- The Sequence Recipient subsequently allocates a SSB at its end, associated with the sequence the Sequence Initiator has initiated.

- Both the Sequence Initiator and Sequence Recipient N_Ports track the status of this Sequence respectively through the Sequence Initiator and the Sequence Recipient SSBs.

The maximum number of concurrent Sequences between two N_Ports is limited to the smaller of the number of SSBs available at these N_Ports. This value is established during N_Port Login through the Service Parameters (see clause 23).

4.13.4 Exchange

An Exchange is composed of one or more nonconcurrent Sequences (see clause 24). An Exchange may be unidirectional or bidirectional. A unidirectional Exchange results when all the Sequences within the Exchange are initiated by the same N_Port. A bidirectional Exchange results when the Sequences within the Exchange are initiated by both the N_Ports, but not concurrently. An Exchang may be performed within one Class 1 Connection or may be continued across multiple Class 1 Connections.

NOTE - FC-PH does not specify an Exchange performed concurrently across multiple N_Ports.

4.13.4.1 Exchange_Identifiers (OX_ID and RX_ID)

Exchange_Identifiers may be used by the Originator and Responder to uniquely identify an Exchange.

- The Originator assigns each new Exchange an Originator Exchange_Identifier (OX_ID) unique to the Originator or Originator-Responder pair and embeds it in all frames of the Exchange.

- The Responder assigns a Responder Exchange_Identifier (RX_ID) unique to the Responder or Responder-Originator pair and communicates it to the Originator before the end of the first Sequence of the Exchange in Classes 1 and 2 (see 24.5). The Responder embeds the RX_ID along with OX_ID in all subsequent frames of the Exchange.

- On receiving the RX_ID from the Responder, the Originator embeds both the RX_ID and OX_ID in all subsequent frames of the Exchange.

The Originator may initiate multiple concurrent Exchanges, but each uses a unique OX_ID. If the X_ID reassignment protocol is used, X_ID value may be reassigned during a given Exchange (see 25.3.1).

4.13.4.2 Association_Header

In some system models, there is a need to associate an Exchange with higher-level system constructs (see clause 25). When an X_ID (OX_ID or RX_ID) is invalidated during an Exchange in such system models, the Association_Header is used to locate the Exchange Status Block (see 25.3.1 and 25.3.2). A specific process or group of processes in such system models, is also identified using the Association_Header (see 25.2). The format of the Association_Header is specified in 19.4.

The support requirements of these two functions are determined during N_Port Login (see 23.6.8.3).

4.13.4.3 Exchange Status Bl cks

An Exchange Status Block is a logical construct representing the format of the Exchange status information (see 24.8). It is used to track the progress of an Exchange on a Sequence by Sequence basis. An Originator Exchange Status Block and a Responder Exchange Status Block respectively are used by an Originator and a Responder to track the status of an Exchange.

– When an Originator initiates an Exchange, it assigns an Originator Exchange Status Block associated with the OX_ID assigned to the Exchange (see 24.8.1).

– The Responder assigns a Responder Exchange Status Block associated with the RX_ID the Responder assigned to the Exchange. The Responder references the Responder Exchange Status Block through its respective RX_ID (see 24.8.1).

– Both the Originator and the Responder track the status of the Exchange at their respective N_Ports.

4.13.5 Protocol

FC-PH provides and describes data transfer protocols and rules to be used by higher level protocols to accomplish a given function. FC-PH provides Login and Logout control functions to manage the operating environment to perform data transfers. FC-PH specifies Primitive Sequences which are low-level Ordered Sets providing word synchronization and a handshake mechanism and ensuring that both the transmitter and receiver are able to transmit and receive Primitive Signals (see 16.4).

– Primitive Sequence protocols: Primitive Sequence protocols are based on Primitive Sequences and specified for Link Failure, Link Initialization, Link Reset, and Online to Offline transition (see 16.6).

– Fabric Login protocol: An N_Port interchanges Service Parameters with the Fabric if present, by explicitly performing the Fabric Login protocol or implicitly through an equivalent method not defined in FC-PH (see 23.1 and 23.3).

– N_Port Login protocol: Before performing data transfer, the N_Port interchanges its Service Parameters (see 23.1, 23.4, and 23.6) with another N_Port explicitly through an N_Port Login protocol or implicitly through an equivalent method not defined in FC-PH.

– Data transfer protocol: The ULP data is transferred using Data transfer protocols. For examples, see annex P.

– N_Port Logout protocol: An N_Port may request removal of its Service Parameters from the other N_Port by performing an N_Port Logout protocol (see 23.5). This request may be used to free up resources at the other N_Port.

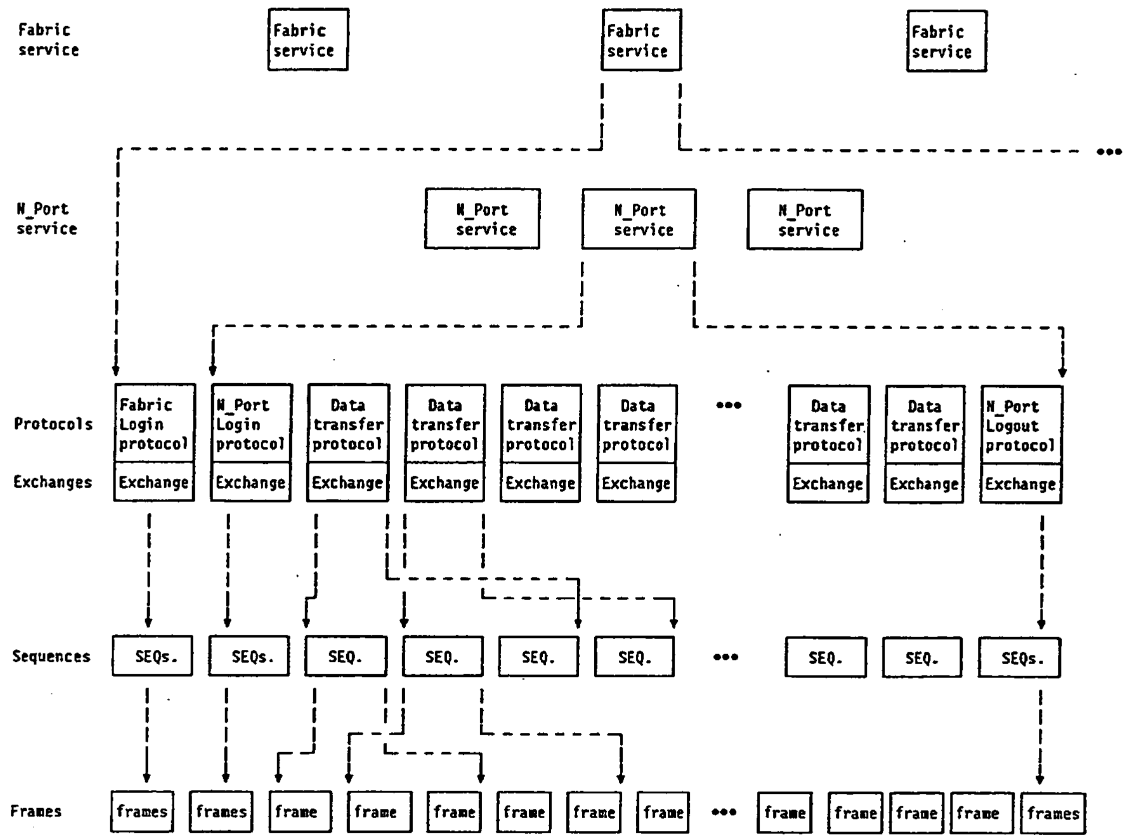


Figure 17 - FC-2 building blocks hierarchy

4.14 Segmentati n and reassembly

Segmentation and reassembly (see clause 27) are the FC-2 functions provided to subdivide application data to be transferred into Payloads, embed each Payload in an individual frame, transfer these frames over the link and reassemble the application data at the receiving end as sent by the sending end.

4.14.1 Application data mapping

Mapping application data to Upper Level Protocols (ULPs) is outside the scope of FC-PH. ULPs maintain the status of application data transferred.

4.14.2 Sending end mapping

Upper levels at the sending end specify to FC-2

- blocks or subblocks to be transferred within a Sequence,
- Information Category for each block or subblock (see 18.2),
- a Relative Offset space starting from zero, representing a ULP-defined origin, to the highest address, for each Information Category (see 27.2.1), and
- an Initial Relative Offset for each block or subblock to be transferred.

The Relative Offset relationship between the blocks to be transferred in multiple Sequences is defined by an upper level and is transparent to FC-2.

Examples of mapping multiple blocks or subblocks are illustrated in figures 18 and 19.

4.14.3 Relative Offset

Relative Offset is an FC-2 construct used to indicate the displacement of the first data byte of each frame's Payload with reference to a ULP-defined origin for a block or a collection of subblocks corresponding to an Information Category at the sending end.

If Relative Offset is not supported (see clause 23), SEQ_CNT is used to perform the segmentation and reassembly (se clause 27).

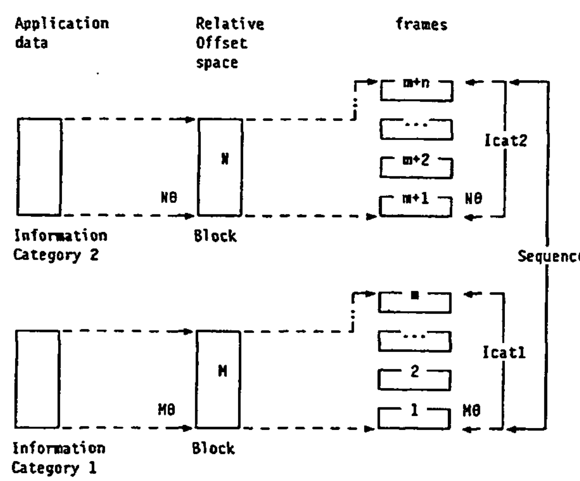
4.14.4 Capability

The Sequence Recipient indicates during Login its capability to support Continuously Increasing or Random Relative Offset (see clause 23). If only the former is supported, each Information Category transferred within a Sequence is treated as a block by upper levels. If Random Relative Offset is supported, an Information Category may be specified as subblocks by upper levels and subblocks may be transmitted in a random order.

4.14.5 FC-2 mapping

The FC-2 maps each block as a single Information Category within the Relative Offset space specified with Continuously Increasing Relative Offset. The FC-2 maps one or more Information Categories into a single Sequence as specified.

The FC-2 maps a collection of subblocks corresponding to an Information Category into a single Sequence within the Relative Offset space as specified with Random Offset.



Legend:

m and n - number of frames for M and N
M0 and N0 - Initial Relative Offset for M and N
Icat1 - Information Category 1
Icat2 - Information Category 2

Figure 18 - Mapping multiple blocks to a single S qu nce

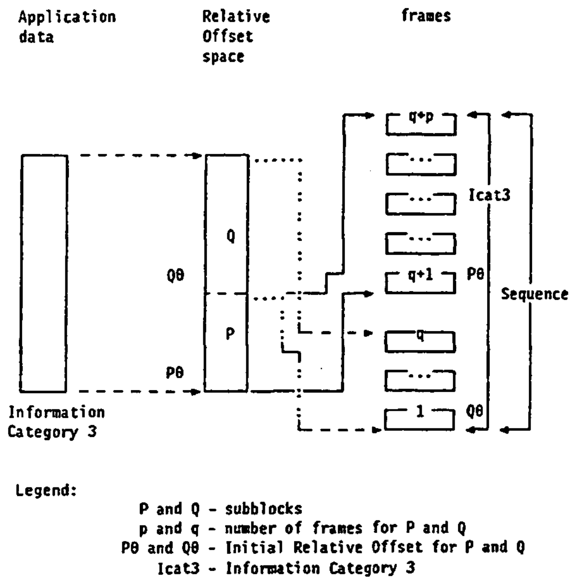


Figure 19 - Mapping subblocks to a single Sequence

4.14.6 Segmentation

Segmentation is the process by which the FC-2 subdivides a block or subblock to be transferred into frames with Payloads of equal or varied sizes and transmits them over the link. During segmentation, the FC-2 determines the size of Payload for each frame in the Sequence. It embeds the data in the Payload of the frame along with the Relative Offset of the Payload.

Relative Offset for the first frame of the block quals the Initial Relative Offset value specified to the FC-2. The Initial Relative Offset specified may be zero or non-zero. Relative Offset for each succeeding frame of the block or subblock is computed as the sum of Relative Offset and the Payload of the current frame.

From the FC-2 perspective, the Relative Offset management is very similar for a block or subblock. Since subblocks may be transferred in a random order, the FC-2 is requested to transfer subblocks, only if the recipient supports Random Relative Offset as specified during Login.

If Relative Offset is Continually Increasing, a block contains only one subblock. If Relative Offset is Random, a block consists of more than one subblock and the upper level specifies the Initial Relative Offset of each subblock; the Initial Relative Offsets of successive blocks are allowed to vary as required by the upper level.

4.14.7 Reassembly

Reassembly is the FC-2 process which reassembles each block or subblock by placing the Payload from each frame at the Relative Offset specified in the frame or by using the SEQ_CNT if Relative Offset is not supported (see clause 27). Reassembly is performed on an Information Category basis.

The FC-2 segmentation and reassembly processes are illustrated in figure 20.

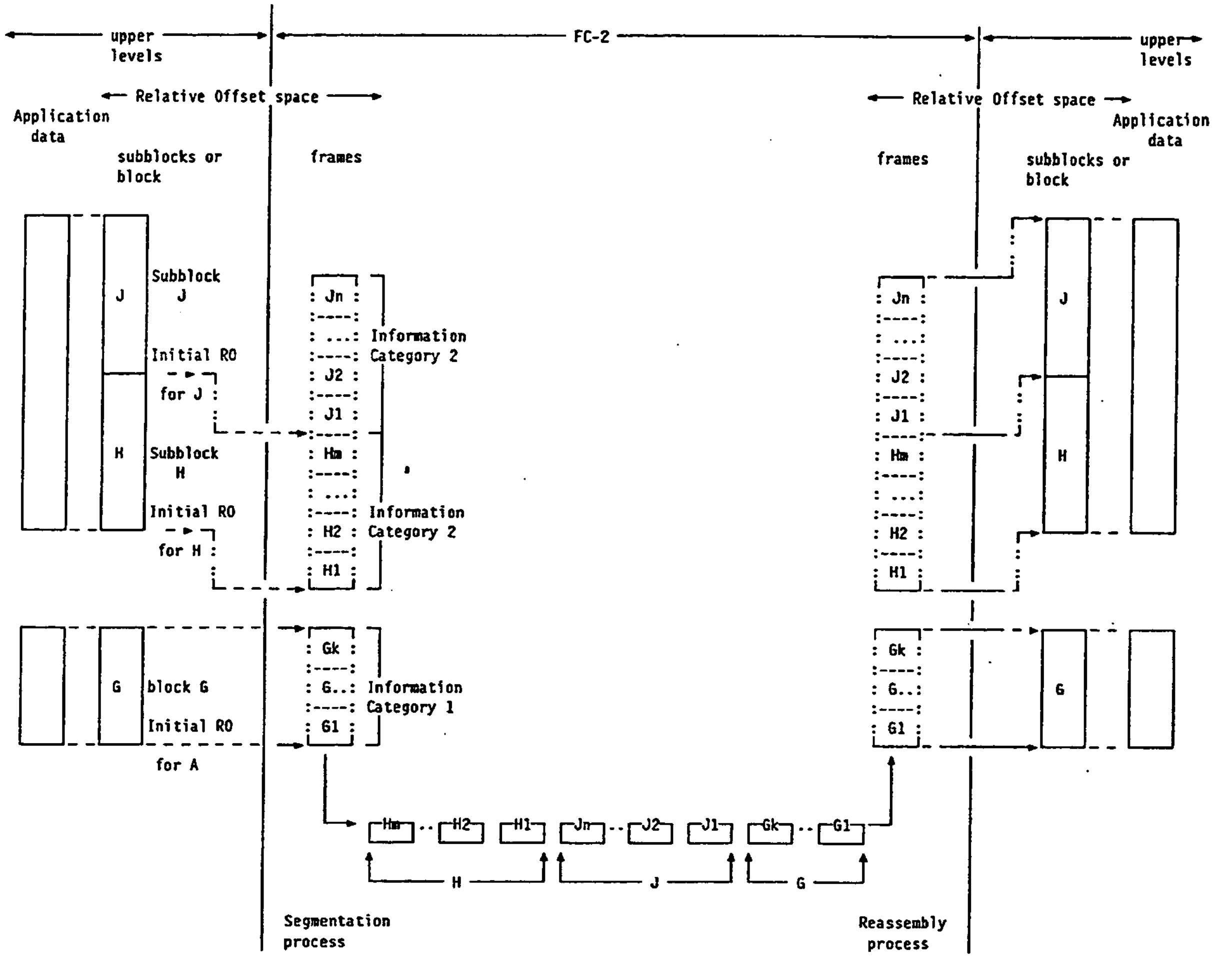


Figure 20 - Segmentation and reassembly

4.15 Error detection and recovery

In general, detected errors fall into two broad categories: frame errors and link-level errors. Frame errors result from missing or corrupted frames. Corrupted frames are discarded and the resulting error is detected at the Sequence level. At the Sequence level, a missing frame is detected or the Sequence times out due to one or more missing Data frames or Acknowledgments. If the discard policy is used, the Sequence is aborted at the Sequence level once an error is detected. Sequence errors may also cause Exchange errors which may also cause the Exchange to be aborted. Error recovery may be performed on the failing Sequence or Exchange with the involvement of the sending ULP. Other properly performing Sequences are unaffected.

Link-level errors result from errors detected at a lower level of granularity than frames, where the basic signal characteristics are in question. Link-level errors include such errors as loss of signal, loss of synchronization and several link timeout errors which indicate no frame activity. Link-level errors may be isolated to a portion of the link. Recovery from link-level errors is accomplished by transmission and reception of Primitive Sequences. Recovery at the link-level disturbs normal frame flow and may introduce Sequence errors which must be resolved after recovery at the link-level.

Err r rec very hierarchy
The recovery of errors may be described by the following hierarchy:

Abort Exchange
(Abort Exchange Protocol-frames)

Abort Sequence
(Abort Sequence Protocol-frames)

Link Reset
(Link Reset Protocol-primitives)

Link Failure
(Link Failure Protocol-primitives)

Link Initialization
(Link Init Protocol-primitives)

5 FC-0 functional characteristics

The FC-0 describes the lowest level physical link in the Fibre Channel system. It is designed for maximum flexibility and allows the use of a large number of technologies to meet the widest variety of system requirements.

5.1 General characteristics

FC-0 has the following general characteristics.

- In the physical media a logical "1" shall be coded as follows:
  a) Optical power - the state with more optical power is a logical "1".
  b) Coaxial - the state where the center conductor (with respect to the shield) is more positive is a logical "1".
  c) Shielded Twisted Pair - the state where the conductor pin identified as "+" is more positive than the conductor pin identified as "-" is a logical "1".
- Serial data streams are supported at data rates of 133, 266, 531 and 1 063 Mbaud (Refer to 3.3 for exact data rates) All data rates have clock tolerances of ±100 ppm.
- A link bit error rate (BER) $\leq 10^{-12}$ is supported. The BER applies to the encoded serial data stream placed on the transmission medium.
- The interoperability specifications are at the external interface as shown at points S and R in figure 8.
- The interface to FC-1 occurs at the logical encoded data interfaces. As these are logical data constructs no physical implementation is implied. However, in a practical transmitter these would, most likely, be represented by the serial data streams. For the transmitter this is point b in figure 9. In the case of the receiver the interface is the retimed serial data indicated at point d in figure 10. These and other points are discussed and recommended in the annexes.

The physical links have the following characteristics:

- Physical point-to-point data links.
- Any transmitter is always connected to the same receiver, optical switches are not supported.
- An elastic buffer is required at all fabric or repeater nodes. All signals shall be retimed before retransmission to prevent jitter accumulation.
- All users are cautioned that detailed specifications shall take into account end-of-life worst case values (i.e. manufacturing, temperature, power supply and cable plant variations).

The interface between FC-0 and FC-1 is intentionally structured to be technology and implementation independent. That is, the same set of commands and services may be used for all signal sources and communications media. The intent is to allow for the interface hardware to be interchangeable in a system without regard to the technology of a particular implementation. As a result of this, all safety or other operational considerations which may be required for a specific communications technology are to be handled by the FC-0 level associated with that technology. An example of this would be the open fibre control system required for the short wave laser technologies (see 6.2.3).

5.2 FC-0 States

5.2.1 Transmitter States

The transmitter is controlled by the FC-1 level. Its function is to convert the serial data received from the FC-1 level into the proper signal types associated with the operating media.

- **Transmitter Not-Enabled State**: A not-enabled state is defined as the optical output off for optical communication media and a logical zero for electrical media. This is the state of the transmitter at the completion of its power on sequence unless the transmitter is specifically directed otherwise by the FC-1 level.
- **Transmitter Enabled State**: The transmitter shall be deemed to be in an enabled state when the transmitter is capable of operation within its specifications while sending valid data patterns.

- **Transition Betwe n Not-Enabl d and Enabled States**: The sequence of events required for the transition between the not-enabled and enabled states are media dependant, both as to the time period required and the optical or electrical activity on the media interface.
- **Transmitter Failure State**: Some types of transmitters are capable of monitoring themselves for internal failures. Examples are laser transmitters where the monitor diode current may be compared against a reference to determine a proper operating point. Other transmitters, such as Light Emitting Diodes and electrical transmitters do not typically have this capability. If the transmitter is capable of performing this monitoring function then detection of a failure shall cause entry into the failure state.

5.2.2 Receiver States

The function of the receiver is to convert the incoming data from the form required by the communications media employed, retime the data, and present the data and an associated clock to the FC-1 level.

The receiver has no states.

5.3 Response to input data phase jumps

Some link_control_facilities may detect phase discontinuities in the incoming data. This may occur from the operation of an asynchronous serial switch at the transmitter. In the event of a phase discontinuity , the recovery characteristics of the receiver shall be as follows:

Phase Jump
Uniform distribution between $\pm 180°$

Link Worst Case

Degree of Recovery
Within BER objective (10^{-12})

Probability of Recovery
95%

Recovery Time
2 500 bit intervals from last phase jump

Additional Wait Time Before N xt Phase Jump
None

The FC-0 level shall require no intervention from higher levels to perform this recovery. If, at the end of the specified time, the higher levels determine that bit synchronization is not present these levels may assume a fault has occurred and take appropriate action.

5.4 Limitations on invalid code

FC-0 does not detect transmit code violations, invalid ordered sets, or any other alterations of the encoded bit stream (see 29.3.2). However, it is recognized that individual implementations may wish to transmit such invalid bit streams to provide diagnostic capability at the higher levels.

Any transmission violation, such as invalid ordered sets, which follow valid character encoding rules as set forth in 11.2 are transparent to FC-0 and will cause no difficulties.

Invalid character encoding could possibly cause a degradation in receiver sensitivity or loss of bit synchronization. To prevent this the transmitted bit stream shall meet the following requirements.

- The maximum code balance in any 10 bits shall lie in the range of 40 % to 60 %. For example the pattern "1010110101" has a code balance of 6/10 = 60 %. The maximum run length is limited to 12 bits in 20 bits. For example the pattern "0011111010" has a run length of 5.
- Between the 20 bit groups from the previous item the transmitted bit stream shall have a contiguous set of at least 300 bits with a balance between 49,5% and 50,5%. The run length in this set shall be limited to 5 bits.

5.5 Receiver initialization time

The time interval required by the receiver from the initial receipt of a valid input to the time that the receiver is synchronized to the bit stream and delivering valid retimed data within the BER objective (10^{-12}), shall not exceed 1 ms. Should the retiming function be implemented in a manner that requires direction from a higher level to start the initialization process, the time interval shall start at the receipt of the initialization request.

5.6 Signal det ct function

The FC-0 may optionally have a signal detect function. If implemented, this function shall indicate when a signal is present at the input to the receiver. Provided the signal detect function does not degrade the sensitivity, otherwise an appropriate guardband shall be included. The activation level shall lie in a range whose upper bound is the minimum specified sensitivity of the receiver and whose lower bound is defined by a complete removal of the input connector.

While there is no defined hysteresis for this function there shall be a single transition between output logic states for any monotonic increase or decrease in the input signal power occurring within the reaction time of the signal detect circuitry. For optical links that employ a link control function, such as the short wavelength Laser Links (see 6.2.3 and annex I), the signal detect function is replaced by the link status function which provides the same service.

The reaction time to the optical power crossing the detect limits shall be less than 12 s.

5.7 FC-0 nomenclature

The nomenclature for the technology options are illustrated in figure 21.

100-SM-LL-L

speed
- 100 100 MB/sec
- 50 50 MB/sec
- 25 25 MB/sec
- 12 12,5 MB/sec

media
- SM single-mode
- M5 multimode (50μm)
- M6 multimode (62,5μm)
- TV video cable
- MI miniature cable
- TP twisted pair

distance
- L long distance
- I intermediate distance
- S short distance

transmitter
- LL long wave laser
- SL short wave laser
- LE long wave LED
- EL electrical

Figure 21 - FC-0 nomenclature

5.8 FC-0 technology options

FC-0 provides for a large variety of technology options. Tables 2 and 3 list the primary signal interface types, which are optical fibre and electrical cable. For each option the nomenclature is listed along with a reference to the clause within this standard containing detailed information.

In tables 4 and 5 the technology options are listed by the communications media types, both optical fibre and electrical cable. Each signal type has an associated primary media type. However, in some cases there is a listing for an alternate cable plant which may be used with degraded performance. These alternate cable plants are indicated by an asterisk in table 4. For more information on alternate cable plants see annex C and annex F.

Table 2 - Optical M dia Signal Interface Overview				
100 MB/sec 1,062 5 Gbaud				
100-SM-LL-L Subclause 6.1 SM 1 300nm 2m-10km	**100-SM-LL-I** Subclause 6.1 SM 1 300nm 2m-2km	**100-M5-SL-I** Subclause 6.2 MM 780nm 2m-500m		
50 MB/sec 531,25 Mbaud				
50-SM-LL-L Subclause 6.1 SM 1 300nm 2m-10km	**50-M5-SL-I** Subclause 6.2 MM 780nm 2m-1km			
25 MB/sec 265,625 Mbaud				
25-SM-LL-L Subclause 6.1 SM 1 300nm 2m-10km	**25-SM-LL-I** Subclause 6.1 SM 1 300nm 2m-2km	**25-M5-SL-I** Subclause 6.2 MM 780nm 2m-2km	**25-M6-LE-I** Subclause 6.3 MM(LED) 1 300nm 2m-1,5km	
12,5 MB/sec 132,812 5 Mbaud				
12-M6-LE-I Subclause 6.3 MM(LED) 1 300nm 2m-1,5km				

Table 3 - Electrical Media Signal Interface Overview				
100 MB/sec 1,062 5 Gbaud				
100-TV-EL-S Subclause 7.2 0-25m	**100-MI-EL-S** Subclause 7.2 0-10m			
50 MB/sec 531,25 Mbaud				
50-TV-EL-S Subclause 7.2 0-50m	**50-MI-EL-S** Subclause 7.2 0-15m			
25 MB/sec 265,625 Mbaud				
25-TV-EL-S Subclause 7.2 0-75m	**25-MI-EL-S** Subclause 7.2 0-25m	**25-TP-EL-S** Subclause 7.3 0-50m		
12,5 MB/sec 132,812 5 Mbaud				
12-TV-EL-S Subclause 7.2 0-100m	**12-MI-EL-S** Subclause 7.2 0-35m	**12-TP-EL-S** Subclause 7.3 0-100m		

Table 4 - Optical Cable Plant Overview				
Single-mode				
100-SM-LL-L Subclause 6.1 SM 1 300nm 2m-10km	**100-SM-LL-I** Subclause 6.1 SM 1 300nm 2m-2km	**50-SM-LL-L** Subclause 6.1 SM 1 300nm 2m-10km	**25-SM-LL-L** Subclause 6.1 SM 1 300nm 2m-10km	**25-SM-LL-I** Subclause 6.1 SM 1 300nm 2m-2km
Multimode (62,5μm)				
100-M6-SL-I ** Subclause 6.2 MM 780nm 2m-175m	**50-M6-SL-I **** Subclause 6.2 MM 780nm 2m-350m	**25-M6-SL-I **** Subclause 6.2 MM 780nm 2m-700m	**25-M6-LE-I** Subclause 6.3 MM(LED) 1 300nm 2m-1,5km	**12-M6-LE-I** Subclause 6.3 MM(LED) 1 300nm 2m-1,5km
Multimode (50μm)				
100-M5-SL-I Subclause 6.2 MM 780nm 2m-500m	**50-M5-SL-I** Subclause 6.2 MM 780nm 2m-1km	**25-M5-SL-I** Subclause 6.2 MM 780nm 2m-2km	**25-M5-LE-I **** Subclause 6 3 MM(LED) 1 300nm	**12-M5-LE-I **** Subclause 6.3 MM(LED) 1 300nm

** alternate fibre cable plant, see annex C.

Table 5 - Electrical Cable Plant Overview				
Video Coax				
100-TV-EL-S Subclause 7.2 0-25m	**50-TV-EL-S** Subclause 7.2 0-50m	**25-TV-EL-S** Subclause 7.2 0-75m	**12-TV-EL-S** Subclause 7.2 0-100m	
Miniature Coax				
100-MI-EL-S Subclause 7.2 0-10m	**50-MI-EL-S** Subclause 7.2 0-15m	**25-MI-EL-S** Subclause 7.2 0-25m	**12-MI-EL-S** Subclause 7.2 0-35m	
Shielded twisted pair				
25-TP-EL-S Subclause 7.3 0-50m	**12-TP-EL-S** Subclause 7.3 0-100m			

6 Optical fibre interface sp cification

This clause defines the optical signal characteristics at the interface connector receptacle. Each conforming optical FC attachment shall be compatible with this optical interface to allow interoperability within an FC environment. Fibre Channel links shall not exceed the BER objective (10^{-12}) under any conditions. The parameters specified in this clause support meeting that requirement under all conditions including the minimum input power level. The corresponding cable plant specifications are described in clause 8.

6.1 SM data links

Table 6 gives the link budgets for 2 and 10 km single-mode optical fibre links running at 266, 531 and 1 063 Mbaud. The optical power coupled into the fibre shall be limited to a maximum value consistent with Class 1 laser safety operation in accordance with IEC 825.

Table 6 - FC-0 physical links for single-mode classes

FC-0	Units	100-SM -LL-L	100-SM -LL-I	50-SM -LL-L	25-SM -LL-L	25-SM -LL-I
Subclause		6.1	6.1	6.1	6.1	6.1
Data Rate	MB/sec	100	100	50	25	25
Nominal Bit Rate	Mbaud	1 062,5	1 062,5	531,25	265,625	265,625
Tolerance	ppm	±100	±100	±100	±100	±100
Operating Range (typ)	m	2-10k	2-2k	2-10k	2-10k	2-2k
Fibre core diameter	μm	9	9	9	9	9
Transmitter (S)						
Type		Laser	Laser	Laser	Laser	Laser
Spectral Centre Wavelength	nm (min.) nm (max.)	1 285 1 330	1 270 1 355	1 270 1 355	1 270 1 355	1 270 1 355
RMS Spectral Width	nm (max.)	3	6	3	6	30
Launched power, max. Launched power, min.	dBm (ave.) dBm (ave.)	-3 -9	-3 -12	-3 -9	-3 -9	-3 -12
Extinction Ratio	dB (min.)	9	9	9	6	6
RIN_{12} (max.)	dB/Hz	-116	-116	-114	-112	-112
Eye Opening @ BER=10^{-12}	% (min.)	57	57	61	63	63
Deterministic Jitter	% (p-p)	20	20	20	20	20
Receiver (R)						
Received power, min.	dBm (ave.)	-25	-20	-25	-25	-20
Received power, max.	dBm (ave.)	-3	-3	-3	-3	-3
Optical path power penalty	dB (max.)	2	2	2	2	2
Return loss of receiver	dB (min.)	12	12	12	12	12

6.1.1 Optical output interface

The general laser transmitter pulse shape characteristics are specified in the form of a mask of the transmitter eye diagram at point S (see figure 8). These characteristics include rise time, fall time, pulse overshoot, pulse undershoot, and ringing, all of which shall be controlled to prevent excessive degradation of the receiver sensitivity. For the purpose of an assessment of the transmit signal, it is important to consider not only the eye opening, but also the overshoot and undershoot limitations. The parameters specifying the mask of the transmitter eye diagram are shown in figure 22.

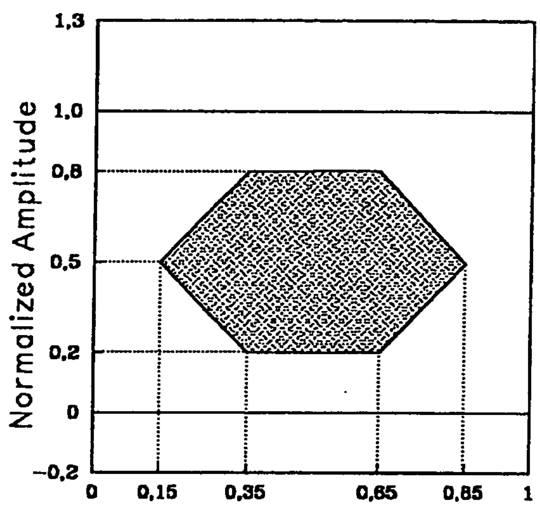


Figure 22 - Transmitter eye diagram mask

The mask of the eye diagram for the laser transmitters shall be measured using a receiver with a fourth-order Bessel-Thompson transfer function given by:

$$H_p = \frac{105}{105 + 105y + 45y^2 + 10y^3 + y^4}$$

With $y = 2.114p$, $p = \frac{j\omega}{\omega_r}$, $\omega_r = 2\pi f_r$

$$f_r = 0{,}75 \times \text{Bit Rate}$$

NOTE - This filter is not int nded to represent the noise filter used within an optical receiver but it is intended to provide a uniform measurement condition.

The nominal attenuation at the reference frequency, f_r, is 3 dB. The corresponding attenuation and group delay distortion at various frequencies is given in table 7.

Table 7 - Transmit pulse noise filter

f/f_0	f/f_r	Attenuation (dB)	Distortion (UI)
0,15	0,2	0,1	0
0,3	0,4	0,4	0
0,45	0,6	1,0	0
0,6	0,8	1,9	0,002
0,75	1,0	3,0	0,008
0,9	1,2	4,5	0,025
1,0	1,33	5,7	0,044
1,05	1,4	6,4	0,055
1,2	1,6	8,5	0,10
1,35	1,8	10,9	0,14
1,5	2,0	13,4	0,19
2,0	2,67	21,5	0,30

Table 8 - Tx Pulse Noise Filter Attenuation Tolerance

Reference Frequency f/f_r	Attenuation Tolerance Δa (dB -typical)
0,1 - 1,00	$\pm 0{,}5$
1,00 ... 2,00	$\pm 0{,}5$ ... $\pm 3{,}0$

Note: Intermediate values of Δa are to be linearly interpolated on a logarithmic frequency scale.

The mask of the eye diagram is intended to measure the waveshape properties of the optical signal such as rise/fall time, overshoot, undershoot, and ringing. As such this test applies to the deterministic portion of the waveform and does not include random noise in either time or amplitude.

The measurement of the extinction ratio shall be made using the methods of OFSTP-4. The measurement may be made with the node transmitting any suitably random valid Fibre Channel traffic.

The optical power measurement shall be made by the methods of OFSTP-2. The measurement may be mad with the node transmitting an idle sequence or other valid Fibre Channel traffic.

6.1.2 Optical input interface

The receiver shall operate within the BER objective (10^{-12}) over the link's lifetime and temperature range when the input power falls in the range given in table 6 and when driven by a data stream output that fits the specified eye diagram mask through a cable plant as specified in clause 8. The measurement shall be made by the methods of OFSTP-3.

Receiver characteristics specified in this document are receiver sensitivity, overload, reflectance, and the allowable optical path power penalty from the combined effects of dispersion, reflections, and jitter.

The minimum and maximum values of the average received power in decibels give the input power range to maintain the required BER (10^{-12}).

These values take into account power penalties caused by the use of a transmitter with worst-case combination of transmitter spectral, extinction ratio, and pulse shape characteristics specified for each application described.

The optical path power penalty (in decibels) accounts for the signal degradation along the optical path from reflections, jitter and the combined effects of dispersion resulting from inter-symbol interference, mode-partition noise. These effective losses are not measurable by the static procedures used to evaluate the cable plant and are accounted for as a separate penalty. In the case of the 100-SM-LL-L class the difference between the minimum transmitter output and the receiver input is (-9 dBm) - (-25 dBm) = 16 dB. Two dB are allocated to the optical path power penalty and the remaining 14 dB are allocated to the cable plant.

Table 9 (Page 1 of 2) - FC-0 physical links for multimode classes

FC-0	Units	100-M5 -SL-I	50-M5 -SL-I	25-M5 -SL-I	25-M6 -LE-I	12-M6 -LE-I
Subclause		6.2	6.2	6.2	6.3	6.3
Data Rate	MB/sec	100	50	25	25	12,5
Nominal Bit Rate	Mbaud	1 062,5	531,25	265,625	265,625	132,812 5
Tolerance	ppm	±100	±100	±100	±100	±100
Operating Range (typ)	m	2-500	2-1k	2-2k	2-1,5k	2-1,5k
Fibre core diameter	μm	50	50	50	62,5	62,5
Transmitter (S)						
Type		Laser	Laser	Laser	LED	LED
Spectral Centre Wavelength	nm (min.) nm (max.)	770 850	770 850	770 850	1 280 1 380	1 270 1 380
Spectral Width	nm RMS (max.) nm FWHM (max.)	4 NA	4 NA	4 NA	NA fig 26	NA 250
Launched power, max. Launched power, min.	dBm (ave.) dBm (ave.)	1,3 -7	1,3 -7	0 -5	-14 -20	-14 -22
Extinction Ratio	dB (min.)	6	6	6	9	10
RIN_{12} (max.)	dB/Hz	-116	-114	-112	NA	NA
Eye Opening @ BER=10^{-12}	% (min.)	57	61	63	NA	NA
Deterministic jitter	% (p-p)	20	20	20	16	24
Random Jitter	% (p-p)	NA	NA	NA	9	12
Optical Rise/Fall Time	ns (max.)	NA	NA	NA	2,0/2,2	4,0

Tabl 9 (Page 2 of 2) - FC-0 physical links for multimode classes						
FC-0	Units	100-M5 -SL-I	50-M5 -SL-I	25-M5 -SL-I	25-M6 -LE-I	12-M6 -LE-I
Receiver (R)						
Received power, min.	dBm (ave.)	-13	-15	-17	-26	-28
Received power, max.	dBm (ave.)	+1,3	+1,3	0	-14	-14
Return loss of receiver	dB (min.)	12	12	12	NA	NA
Deterministic jitter	% (p-p)	NA	NA	NA	19	24
Random Jitter	% (p-p)	NA	NA	NA	9	12
Optical Rise/Fall Time	ns (max.)	NA	NA	NA	2,5	4,3

6.2 SW laser data links

The first three columns of table 9 give the link budgets, transmitter specifications, and receiver specifications for the short wavelength (SW) laser links operating at the 1063, 531 and 266 Mbaud rates.

6.2.1 Optical output interface

The characteristics of the transmitted signal at the optical output interface are given in table 9. The SW laser described for these links shall have properties that significantly reduce the noise problems associated with using lasers on multimode fibre. In particular, the laser shall be chosen to minimize the effects of modal noise which may occur if the cable plant contains mode selective loss (e.g. poor connections).

Modal noise is worst when the fibre dispersion, between the transmitter and the location of mode selective loss, occurs in an initial length of fibre folowing the transmitter. A common technique for reducing modal noise is to use lasers with short coherence lengths. This in turn reduces the initial length of fibre in which modal noise is most problematic. Within this initial length of fibre, extra care shall be taken to avoid mode selective loss (see clause 8.3.3). One type of short coherence length laser that minimizes the problems of modal noise and laser reflection noise on multi-mode fibre is the self-pulsating type of laser. Other laser oscillation techniques that shorten the laser coherence length include relaxation oscillations and superimposing an external high frequency oscillating drive signal.

The self-pulsation frequency shall be greater than three times the bit rate of th link to allow for efficient filtering of the self-pulsation noise.

The self-pulsation frequency of most self-pulsating lasers shifts upward over a limited range with increased drive current. Hence, if a self-pulsating laser is used for either a 531 or 1062 Mbaud data link, it may be desirable to operate the SW laser at a slightly higher power level than that used for a 266 Mbaud link.

The maximum and minimum of the allowed range of average transmitter power coupled into the fibre are worst-case values to account for manufacturing variances, drift due to temperature variations, aging effects, and operation within the specified minimum value of the extinction ratio.

The minimum value of the extinction ratio shall be the minimum acceptable value of the ratio (in decibels) of the average optical energy in a logic one level to the average optical energy in a logic zero level. The extinction ratio shall be measured under fully modulated conditions in the presence of worst-case reflections.

The transmitted signal must not only meet eye opening requirements, but also overshoot and undershoot limitations. The parameters specifying the (normalized) mask of the transmitter eye diagram are identical to those presented in 6.1.1 and shown in figure 22.

6.2.2 Optical input interface

The receiver shall operate at a BER of 10^{-12} over the link's lifetime and temperature range when the input power falls within the range given in table 9 and when driven by a data stream output that fits the specified eye diagram mask through a cable plant as specified in clause 8.

Receiver characteristics specified in this document are receiver sensitivity, overload, reflectance, and the allowable power penalty from the combined effects of dispersion, reflections, and jitter.

The minimum and maximum values of the average received power in dBm give the input power range to maintain a BER $\leq 10^{-12}$. These values take into account power penalties caused by the use of a transmitter with a worst-case combination of transmitter spectral, extinction ratio, and pulse shape characteristics specified for each version.

The optical path power penalty (in decibels) typically accounts for the total degradation along the optical path resulting from reflections, jitter, and the combined effects of dispersion from intersymbol interference, mode-partition noise, and laser chirp. However, it is a common practice for multi-mode fibre data links, to include this optical path power penalty into the actual link budget specifications of both the power budget (producing amplitude degradation) and the timing budget (producing pulse spreading degradation). Therefore, the SW laser data links have no specified optical path power penalty (in table 9) because the related link degradations of both amplitude and pulse spreading (primarily modal dispersion) are already accounted for in the power budget and time budget as specified between the transmitter and receiver in table 9.

A unique power penalty consideration for SW laser data links on multimode fibre is mode selective loss. Refer to annex E for design considerations related to mode selective loss.

The minimum acceptable value for receiver sensitivity shall equal the values specified in table 9. In addition, the receiver shall be designed to accommodate the maximum received power and the optical rise/fall time at the input to the receiver.

6.2.3 The Open Fibre Control (OFC) Safety System

An overview of the Open Fibre Control (OFC) safety system is present d in this clause. The OFC system can be used as a safety interlock for point-to-point optical fibre links that use semiconductor laser diodes as the optical source. A safety interlock system is necessary for SW laser data links since the optical power levels required to obtain the desired level of system performance exceed the Safety Class 1 limits defined by one or more national or international laser safety standards. Meeting the requirements for Safety Class 1 classification is very important for an optical data-communications system since many access points may exist in unrestricted locations (i.e. offices). Certifying the laser system at a higher classification level is impractical unless access can be restricted to only trained service personnel.

The following description of the OFC system applies to the 1062, 531 and 266 Mbaud SW laser data links. The OFC timings for the 1062 and 531 Mbaud links are different than the timings for the 266 Mbaud link due to the increase in allowed optical power for the 1062 and 531 Mbaud links versus that for the 266 Mbaud link.

6.2.3.1 Operation of the OFC system

During normal operation the optical link is a closed system (i.e., no laser emissions) and therefore Safety Class 1 regardless of the power in the fibre. It is only when the link is opened that a person can be exposed to laser radiation. Hence, one can design a Safety Class 1 system by implementing a safety interlock that can detect when the optical link has been interrupted and shut down the laser or reduce the amount of emitted laser radiation. A block diagram of a point-to-point optical link with OFC circuitry is shown in figure 23 and referenced in the following description of the OFC system.

Depending upon the state of the OFC system, an optical link will either be in an active mode meaning normal data transmission or an inactive mode meaning the link is open or disconnected at some point or a fault condition exists. In addition to a maximum average optical power level for the open transmitter port, there are four timing values used to specify the OFC system:

- Turn-off time, T_{off}, is the maximum time allowed for both ports in a link to switch from active mode (i.e., transmitting data) to inactive mode (i.e., transmitter off) once a link disconnection or fault condition occurs.
- Pulse repetition time, T, defines the frequency at which th OFC system will check the status

of the optical link while in the inactive mode of operation.

- Pulse duration, t, is the time period that the port's transmitter is activated during an OFC link status check or reconnection handshake.
- Stop time, t_s, is the time period that the port's transmitter is deactivated during an OFC link reconnection handshake.

The timing values are specified for each of the SW data links in 6.2.3.3.

The sequence of events which follow a disconnection in the optical data link are as follows:

a) Suppose data link A to B (1) is disconnected; e.g , an opened connector or cut fibre.

b) The receiver on optical port B (2) is no longer receiving an optical signal and triggers a loss-of-light flag to be raised in the control circuitry (3) on port B.

c) The control circuitry (3) responds by forcing off the port B transmitter (4) and starting an internal T second timer.

d) Since the port B transmitter is now off, no optical signal is detected by the port A receiver (6). This triggers a loss-of-light flag to be raised in the control circuitry (7) on port A.

e) The control circuitry (7) responds by forcing the port A transmitter off (8) and starting an internal T second timer. Thus, a safe condition with respect to the opened end of the link (1) has been created since both transmitters are now off.

f) When the internal T second timers expire, the control circuitry on each port shall turn on the port's transmitter for t μs to check the link status; i.e., check for a closed link.

g) If the link is now a closed loop (i.e., data link A to B (1) has been reconnected), then a reconnect handshake takes place between the two ports and the transmitters return to normal operation. If the link is still open, no optical signal is detected by at least one of the two receivers, the reconnect handshake fails and the transmitters are once again turned off for T seconds before the link check is repeated.

h) Note that to allow for synchronization of the control circuitry on the two ports during the link status check and reconnect handshake, either the expiring of the port's internal timer or the receiving of an optical signal from the other port triggers an attempt to reconnect and the port T second timer to be reset.

i) If both data links A to B (1) and B to A (5) are disconnected at the same time, both ports A and B shall independently turn off their transmitters since a loss-of-light signal is generated at each receiver. Normal operation shall not return until both data links are reconnected and the proper reconnect handshake has taken place between the ports.

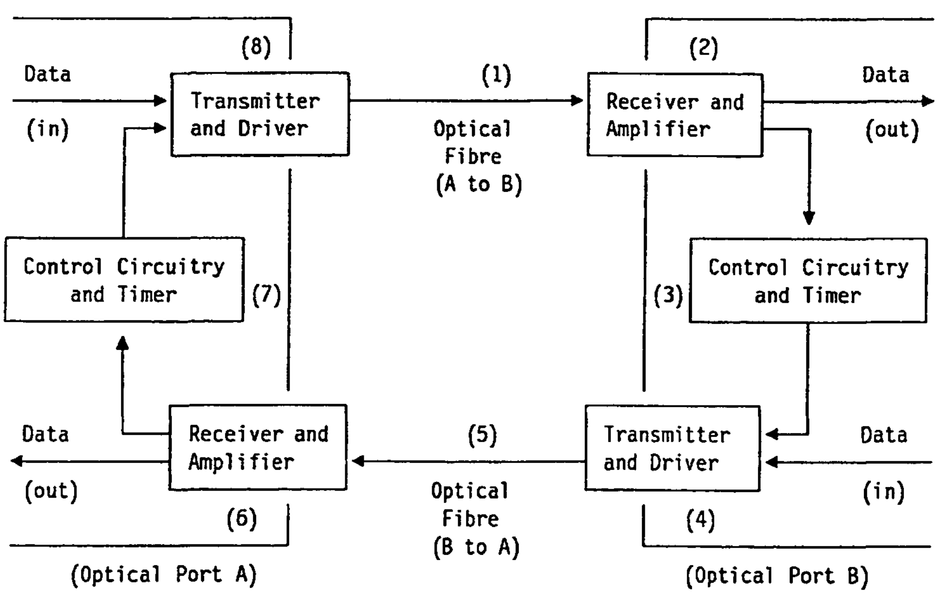


Figure 23 - Block diagram of the OFC safety system

Turning the transmitters on for a brief link check after a predetermined time (T) allows the system to return to a normal mode of operation after an accidental or purposeful disconnection and reconnection of one or more of the connectors. Hence, the timed link checking and reconnection make the OFC safety interlock system automatic without the need for user interaction.

6.2.3.2 Link reconnection algorithm

During system start-up or link reconnection, a handshaking takes place between the two optical ports. This handshaking occurs after a closed link condition is detected and ensures that both optical ports contain functioning safety control circuitry and are capable of shutting down in the event of a break in the link. If either port in the link (e.g., port B) does not respond to the handshaking, then the control circuitry at the other end of the link (port A) keeps its transmitter inactive (i.e., either no emission or brief pulses every T seconds) and thereby maintains a safe link. Hence, the electronic safety control functions as a safety interlock which can not be defeated by attaching an optical port without OFC safety control circuitry.

The reconnection procedure is implemented with an on-off-on algorithm and four-state state machine. The four states are referred to as the Active, Disconnect, Stop and Reconnect States. Each of the states have as input an internal loss-of-light (LOL) signal. This LOL signal shall be the output of a signal detection and digital filtering process that integrates the received signal to improve its reliability (see also 6.2.3.4). The digital filter shall sample at a faster rate during the reconnect sequence than during normal operation when asserting LOL and signalling a link disconnection. The following list describe the function of each of the states of the state machine:

a) The OFC system is in the Active State during normal link operation. In this state, the port's receiver shall be continuously monitored for a loss-of-light (LOL) condition. If a LOL condition is detected (LOL asserted), the OFC system shall transfer control to the Disconnect State.

b) Th Disconnect State is used to maintain a safe (Safety Class 1) link and to repetitively check the link for a closed link condition. Every T seconds the port's transmitter shall be activated for a t μs check period (called decode 1), and its receiver monitored for the LOL condition to disappear (LOL deasserted). The OFC system shall remain in this state until an optical signal is both sent and received during a decode 1 check period. This sending and receiving of light can occur in one of two ways:

   1) The port's T second timer expires, its transmitter is activated for a t μs check period (decode 1), and during this check period, a response is received from the other end of the link; i.e., LOL deasserted. The port shall then be considered "master" of the reconnection attempt and control transferred to the Stop State. (NOTE: Due to timing variations, it is possible for both ends of the link to be "master" at the same time; however,this situation does not interfere with proper functioning of the OFC system).

   2) A light signal is received from the other end of the link sometime during a T second wait period; i.e., LOL deasserted. The port's T second timer shall be reset and its transmitter activated for a t μs check period (decode 1) in order to send a response. In this situation, the port shall be considered "slave" of the reconnection attempt and control is transferred to the Stop State.

c) The Stop State is the "off" portion of the handshake algorithm. The Stop State check period (called decode 2 and specified as 2t or 4t μs) shall not begin until after decode 1 has expired. When decode 2 begins, the port's transmitter shall be disabled and the link monitored for a LOL condition (LOL asserted). The OFC system shall remain in the Stop State until a LOL condition exists, conceivably for an indefinite period of time. The system shall exit from the Stop State in one of two ways depending upon whether the LOL condition occurs during the decode 2 period or after it has expired:

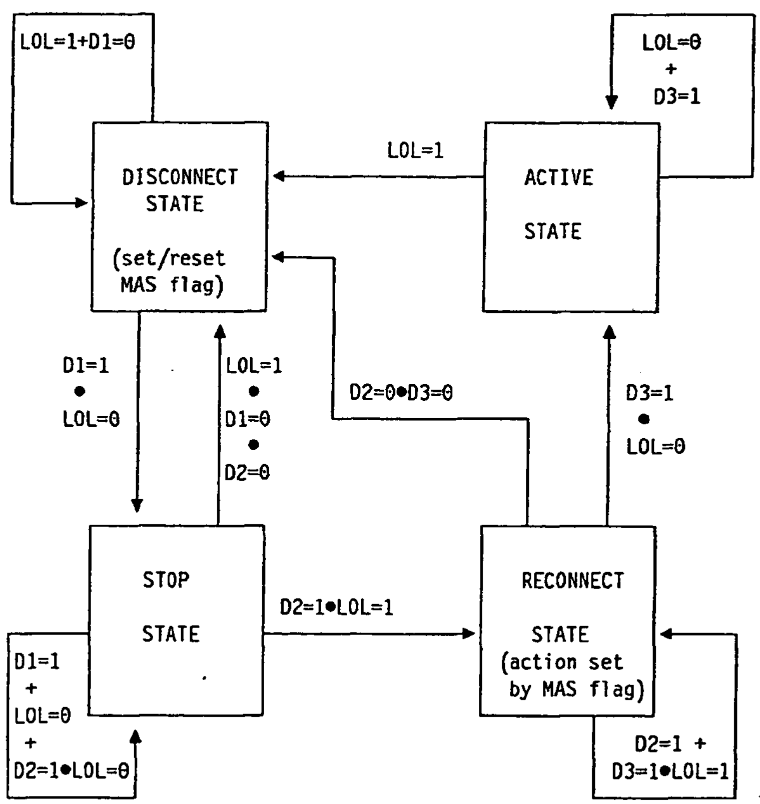
   1) If the LOL condition (LOL asserted) is detected during the decode 2 period, then the OFC system shall transfer control to the Reconnect State.

   2) If the LOL condition (LOL asserted) is detect d after the decode 2 period has expired, then control shall be transferred back to the Disconnect State.

d) The Reconnect State verifies that the link between the two ports is complete and uninterrupted (i.e., a closed link). The verification shall require that an optical signal be both sent and received during a t μs check period (called decode 3). The decode 3 verification process shall not begin until after the decode 2 period has expired. The sequence of events in this state is different depending upon whether the port is "master" or "slave" of the reconnection attempt.

1) If the port is "master" of the reconnection attempt, then it shall again initiate the send/receive sequence by activating its transmitter for a t μs check period (decode 3) and simultaneously monitor its receiver for a response signal (LOL deasserted). If it receives a response within the decode 3 period, it shall keep its transmitter activated and transfer control to the Active State; otherwise, it shall disable the transmitter and transfer control back to the Disconnect State.

2) If the port is "slave" of the reconnection attempt, then it shall monitor its receiver during a t μs check period (decode 3) for the presence of an optical signal (LOL deasserted) but not simultaneously activate its transmitter. If it receives a light present signal within the decode 3 period, then it shall activate its transmitter in order to send a response and transfer control to the Active State; otherwise, it shall keep its transmitter disabled and transfer control back to the Disconnect State.




```
DEFINITIONS:
LOL = Loss of light flag (asserted = 1, deasserted = 0)
MAS = Master of link reconnection flag
D1  = Decode 1 - 1st time period flag  = Link check
D2  = Decode 2 - 2nd time period flag  = Disable laser
D3  = Decode 3 - 3rd time period flag  = Link check
```

Figure 24 - Open Fibre Control modul State Diagram

The timing periods used for the various states of the state machine depend principally upon the optical link length, the port's detection and laser turn-on delays, the safety standards and the emission level for the product. The t μs time period shall be greater than one "worst case" round trip time and yet be sufficiently short to meet Safety Class 1 restrictions for the "worst case" emission level. In addition, note that a successful link reconnection will always require the same amount of time to complete: that is, decode 1 + decode 2 + decode 3. For additional information on the OFC system and state machine, refer to annex I.

6.2.3.3 OFC power values and time periods

The OFC system uses a repetitive pulsing technique (i.e., laser activated for t μs every T seconds) during the time that a link is open in order to reduce the maximum possible exposure to a value which allows for classification as a Safety Class 1 laser product. The maximum average power level per pulse is a function of the wavelength, pulse duration (t), and pulse repetition frequency (PRF = 1/T). The PRF determines the number of pulses (N) that occur during the time base used for classification. It is important to note that the use of the word "pulse" refers to the time (t) during which the laser is activated and being modulated with a valid full rate data pattern. The maximum (worst case) average transmitter output power for the SW laser versions shall be:

Average transmitter receptacle power (maximum):

- 25-M5-SL-I: 1,7 dBm (1,48 mW)
- 50-M5-SL-I: 3,2 dBm (2,10 mW)
- 100-M5-SL-I: 3,2 dBm (2,10 mW)

To function correctly, each SW optical data link port must contain a transmitter/receiver unit that has implemented the OFC system with compatible OFC interface timings. The OFC timing values that shall be used for the SW laser data link versions are specified below. They are based upon the maximum transmitter receptacle power values and both national and international laser safety exposure limits for a Safety Class 1 SW laser system.

a) Pulse repetition time, T, (Disconnect State)

- 25-M5-SL-I: 10,1 s (2^{28} τ_{25})
- 50-M5-SL-I: 10,1 s (2^{29} τ_{50})
- 100-M5-SL-I: 10,1 s (2^{30} τ_{100})

b) Pulse duration, t, (decode 1 time period in Disconnect State and decode 3 time period in Reconnect State)

- 25-M5-SL-I: 617 μs (2^{14} τ_{25})
- 50-M5-SL-I: 154 μs (2^{13} τ_{50})
- 100-M5-SL-I: 154 μs (2^{14} τ_{100})

c) Stop time, (decode 2 time period in Stop State)

- 25-M5-SL-I: 1 234 μs (2^{15} τ_{25})
- 50-M5-SL-I: 617 μs (2^{15} τ_{50})
- 100-M5-SL-S: 617 μs (2^{16} τ_{100})

where τ_{25}, τ_{50} and τ_{100} are the (10 bit) byte times for the respective links and the tolerance in the timings is that which arises from the bit rate tolerance.

While in the inactive mode of operation, the maximum time for the port to respond to a "light present" or "loss-of-light" signal at its receiver is limited by the pulse duration, since the worst case round trip time must be less than this value. This includes any delays from light detection, filtering, the state machine, laser turn on or off, and fibre transmission.

While in the active mode of operation (i.e., normal transmission), the turn-off time, T_{off}, specifies the maximum time between the instant that the data link is disrupted or a fault condition is detected to the instant that laser light is no longer emitted from any point of access in the data link (i.e., each port has switched to inactive mode operation). This turn-off time is longer than the pulse duration or stop times to allow for improved filtering and reliability of the loss-of-light signal. It is specified as follows:

Active mode link turn-off time, T_{off}, (maximum):

- 25-M5-SL-I: 4.0 ms
- 50-M5-SL-I: 2.0 ms
- 100-M5-SL-I: 2.0 ms

These time periods, when used according to th OFC interface specification described in this clause, shall result in a laser product that is under the exposure limits for Safety Class 1 classification as determined by both national and

international laser safety standards in existence in 1991. Note however, that classification of a laser product shall always relate to the standards or regulations in existence at the time of manufacture and shall be supported with optical power measurements and variability analysis (refer to annex I for an example safety calculation).

6.2.3.4 Safety redundancy

The output from a Safety Class 1 laser product shall not exceed the maximum permissible exposure limits under any condition of operation, including single fault conditions. This requirement can be met by introducing redundancy into the OFC system. The OFC system described above shall be implemented using two control paths that provide a redundant means for deactivating the laser drive. The laser drive shall be active only when both control paths agree that that the link is complete; either of the paths shall independently be capable of deactivating the laser drive.

A block diagram of the OFC system with redundant control paths is shown in figure 25. Each path consists of a loss-of-light detector, digital filter, state machine, timers (counters) and a laser driver control line. The two loss-of-light detectors monitor the receiver's signal; at least one of the detectors shall be AC coupled and therefore respond only to a modulated signal.

The two loss-of-light detectors each feed a digital filter; the output of each filter shall be OR/EQUAL'd to form an internal Loss-of-Light (LOL) signal. In the Disconnect, Stop and Reconnect States the "EQUAL" function shall be implemented. The internal LOL signal shall be allowed to toggle from asserted to deasserted or vice versa only when the two filter outputs agree. Once the system is in the Active State, the "OR" function shall be implemented so that only one light detector is required to assert LOL and deactivate the laser.

The internal LOL signals shall be used to synchronize the counters and state machines. The counters shall control the pulse repetition, pulse duration and stop times. The counters shall also provide the low frequency sampling clock to the digital filters. The state machines shall control the handshake algorithm implemented in the OFC system and independent "Laser Off" output lines. Each "Laser Off" output line shall independently be capable of disabling the laser drive circuits via separate control paths.

In addition to the OFC path redundancy, caution shall be used in designing the LOL detector, state machine and laser driver blocks shown in figure 25. If a fault in the circuitry of one of these blocks has a reasonably foreseeable chance of causing other faults, the end result shall be one in which the laser is still deactivated when the link path is disrupted (i.e., opened).

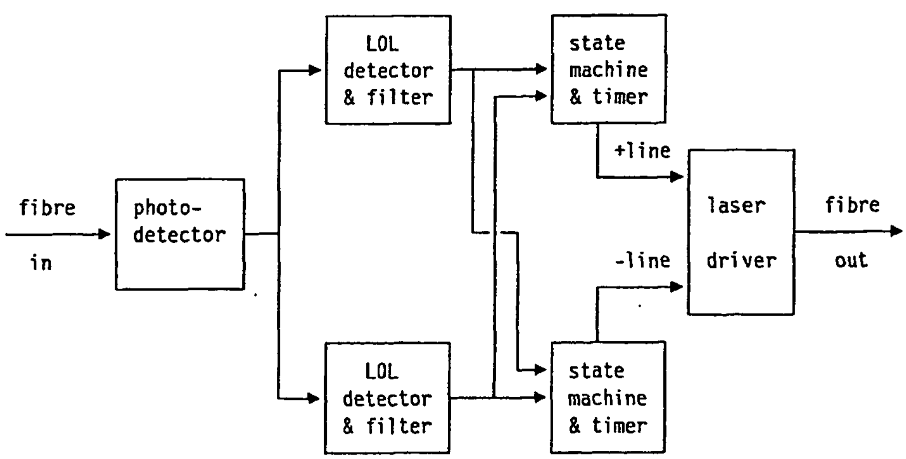


Figure 25 - Bl ck diagram f the OFC safety redundancy

6.2.3.5 Safety docum ntation and usage restrictions

All laser safety standards and regulations require that the manufacturer of a laser product provide information about a product's laser, safety features, labeling, use, maintenance and service. As part of this documentation, the following two usage restrictions shall be included with all shortwave (SW) laser transceiver products utilizing the OFC safety system:

a) The laser product shall be used in point-to-point optical links only. The OFC safety system is incompatible with other types of link connections (i.e., multiple input or output links). Failure to comply with this usage restriction may result in incorrect operation of the link and points of access that may emit laser radiation above the limit for Safety Class 1 systems established by one or more national or international laser safety standards.

b) Normal operation of the point-to-point optical link requires that the laser product shall be connected only to another Fibre Channel compatible laser product that includes the OFC safety system. In addition, each of the these products must be certified as Safety Class 1 laser products according to the laser safety regulations and/or standards in existence at the time of manufacture.

The certification ensures that each of the products will function correctly in the event of a fault in one of the safety control systems.

6.3 LED data links

6.3.1 Optical output interface

The optical output interface shall exhibit characteristics as shown in table 9, when tested according to the methods of annex A.1.2.3

6.3.2 Spectral width

The spectral width is in table 9.

The maximum allowable spectral width is a function of source centre wavelength and source specifications in conjunction with the fibre's chromatic dispersion and modal bandwidth parameters given in clause 8 result in an optical rise time of less than 2,5 ns exiting a 1,5 kilometer fibre cable. Figure 26 shows the permissible spectral width as a function of centre wavelength.

6.3.3 Optical input interface

The optical input interface shall operate when provided with a signal corresponding to table 9 through a cable plant as specified in clause 8.

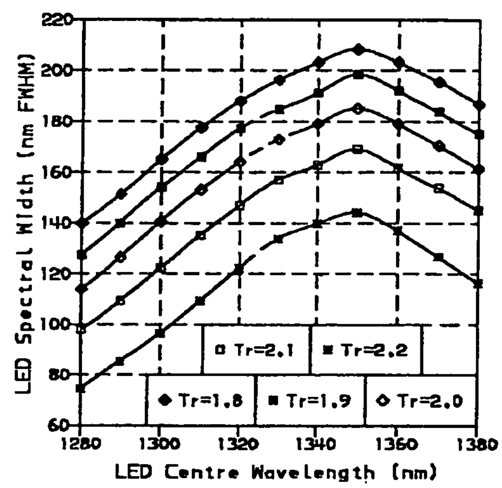


Figure 26 - LED spectral width

7 Electrical cable interface specification

This clause defines the interfaces of the serial electrical signal at the interconnect receptacles. Each conforming electrical FC attachment shall be compatible with this serial electrical interface to allow interoperability within an FC environment. All Fibre Channel links described in this clause shall operate within the BER objectives (10^{-12}). The parameters specified in this clause support meeting that requirement under all conditions including the minimum input power level. The corresponding cable plant specifications are described in clause 9.

The link distance capability specified in Table 10 is based on insuring interoperability across multiple vendors supplying the technologies (both link transceivers and cable plants) under the tolerance limits specified in the document. Greater link distances may be obtained by specifically engineering a link based on knowledge of the technology characteristics and the conditions under which the link is installed and operated. However, such link distance extensions are outside the scope of this standard.

The user needs to take care that their use conditions at least conform to the specified conditions of the document.

Table 10 (Page 1 of 2) - FC-0 physical links for electrical cable classes

FC-0	Units	100-TV-EL-S 100-MI-EL-S	50-TV-EL-S 50-MI-EL-S	25-TV-EL-S 25-MI-EL-S 25-TP-EL-S	12-TV-EL-S 12-MI-EL-S 12-TP-EL-S
Data Rate	MB/s	100	50	25	12,5
Nominal Bit Rate	Mbaud	1 062,5	531,25	265,625	132,812 5
Tolerance	ppm	±100	±100	±100	±100
Operating Distance					
Video	meters	25	50	75	100
Mini-coax	meters	10	15	25	35
Twisted pair	meters	NA	NA	50	100
Cable impedance					
Video	Ω (nom.)	75	75	75	75
Mini-coax	Ω (nom.)	75	75	75	75
Twisted pair	Ω (nom.)	150	150	150	150
S_{11} @ S (.1 to 1.0 bit rate)					
Video	dB	-15	-15	-15	-15
Mini-coax	dB	-7	- 7	- 7	- 7
Twisted pair	dB	NA	NA	-12	-12
Transmitter (S)	Output characteristics at the cable connector				
Type		ECL/PECL	ECL/PECL	ECL/PECL	ECL/PECL
Output Voltage					
- maximum	mV (p-p)	1 600	1 600	1 600	1 600
- minimum	mV (p-p)	600	600	600	600
Deterministic Jitter	% (p-p)	10	10	10	10
Random Jitter	% (p-p)	12	12	8	8
Rise/Fall Time (20-80%)	ns (maximum)	0,4	0,6	1,2	1,8

Table 10 (Page 2 of 2) - FC-0 physical links for electrical cable classes					
FC-0	Units	100-TV-EL-S 100-MI-EL-S	50-TV-EL-S 50-MI-EL-S	25-TV-EL-S 25-MI-EL-S 25-TP-EL-S	12-TV-EL-S 12-MI-EL-S 12-TP-EL-S
Receiver (R)	Input characteristics at the cable connector				
Min Sensitivity (D21.5 idle pattern)	mV	200	200	200	200
Maximum input voltage	mV, p-p	1 600	1 600	1 600	1 600
S_{11} (0,1 to 1,0 bit rate)	dB	-17	-17	-17	-17
Min Discrete Connector Return Loss (.3MHz-1GHz)					
Video	dB	20	20	20	20
Mini-coax	dB	15	15	15	15
Twisted pair	dB	NA	NA	12	12

7.1 Electrical ECL data links

The electrical ECL data link definitions apply to two styles of coaxial cable, the 75 Ω video style coax and miniature style coaxial, and the Shielded Twisted Pair (STP) cable. The electrical characteristics of these links are defined in table 10. The video style coax and miniature coaxial cables are interoperable, i.e., electrically compatible with minor impact on link length capability when intermixed. The interoperability implies that the transmitter and receiver level specifications as measured in figure &cxtx. and defined in table 10 are preserved with the trade-off being distance capability in an intermixed system. Other electrically compatible, interoperable coaxial cables may be used to achieve goals of longer distance, higher frequency, or lower cost as desired in the system implementation provided that they are connector compatible.

The STP cables are incompatible with coaxial cables in terms of characteristic impedance, mode of connection to the transceiver, and other electrical and mechanical parameters. A different connector is specified for STP interface in order to avoid user mixing of coaxial and STP cables. Schematics in the diagrams that follow are for illustration and do not represent the only feasible implementation. The systems below are to be applied only to homogenous ground system applications such as between devices, within a cabinet or rack, or between cabinets interconnected by a common ground return or ground plane. This restriction minimizes safety and interference concerns caused by the voltage differences between equipment grounds.

The recommended interface to the STP and coax is via transformer or capacitive coupling.

7.2 75 Ω coaxial data links

7.2.1 ECL compatible driver characteristics

The output driver is assumed to have Emitter Coupled Logic (ECL) output levels. The output driver shall have output levels, measured at the input to the coax (point S), as shown in table 10, when terminated as shown in figure 28. Measurements are to be made using an oscilloscope

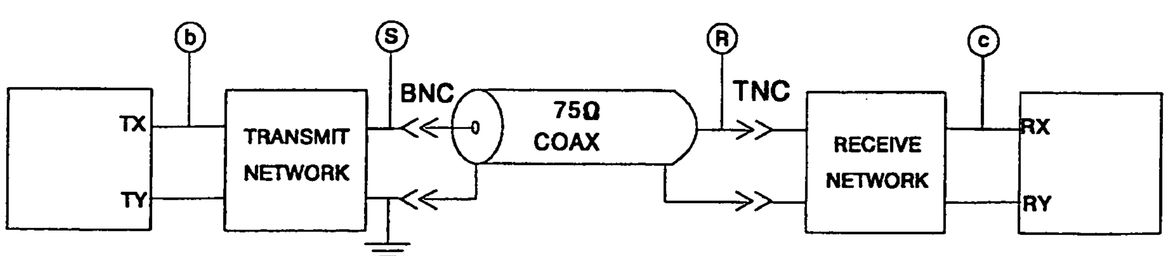


Figure 27 - FC-0 with 75Ω coaxial cable

whose bandwidth including probes is at least 1,8 times the baud rate.

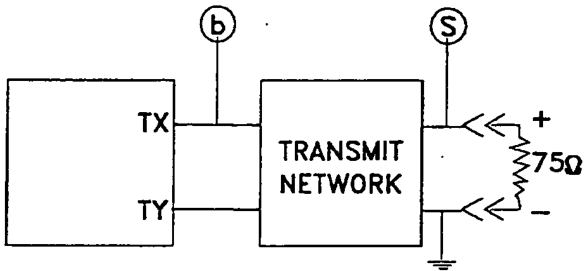


Figure 28 - Coaxial transmitter test circuit

The normalized mask of the transmitter eye diagram is given in figure 29. The transmitter output is specified in table 11.

Table 11 - Transmitter eye diagram mask

Rates	X1	X2	Y1
132,812 5 Mbaud	0,15	0,35	0,30
265,625 Mbaud	0,15	0,35	0,30
531,25 Mbaud	0,15	0,35	0,30
1 062,5 Mbaud	0,15	0,30	0,30

7.2.2 ECL compatible receiver characteristics

At the receiver end the cable is terminated by an equivalent impedance of 75 Ω

An optional equalizer network corrects for timing loss variations due to the differences in propagation delay time between higher frequency components as well as attenuation loss of the transmitted signal. An equalizer design should have fixed values and need no adjustment relative to coaxial line length or levels.

A more detailed discussion of the equalizer is found in annex F.8.

The receiver shall operate within the BER objective (10^{-12}) when a coaxial data link is driven by a transmitter meeting the requirements defined in table 10.

The receiver characteristics of minimum sensitivity and coaxial data link characteristics of maximum path penalty, S_{11} of the rec iver, and minimum discrete connector return loss ar specified in table 10.

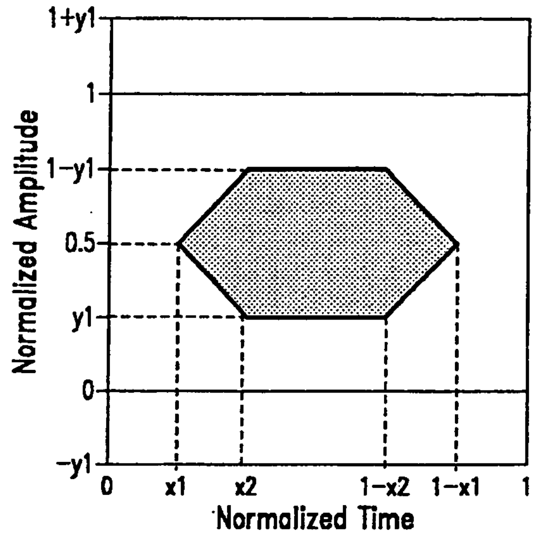


Figure 29 - Transmitter eye diagram mask

7.2.3 Coaxial cable characteristics

The 75Ω coaxial data links may be interconnected using either video or miniature coax depending on the performance and distance required for a specific application.

The coaxial cables are to be grounded only on the transmitter end as shown in figure 27.

7.2.4 Coaxial connectors

7.2.4.1 Connectors for video style coaxial cable

The video style coaxial cable data links are connected using industry standard 75Ω BNC and TNC type connectors as shown in figure 27. The electrical performance of the 75Ω BNC and TNC connectors shall be compatible with video style connectors specified by EIA-403-A. The mechanical compatibility for BNC type (bayonet lock coupling) connectors is defined by MIL-C 39012 which intermates with comparable BNC UG/U connectors.

The mechanical compatibility for TNC type (threaded coupling) connectors is defined by MIL-C 23329 which intermates with comparable TNC UG/U connectors. Primary use of the video style coaxial cable is for interconnection of cabinets.

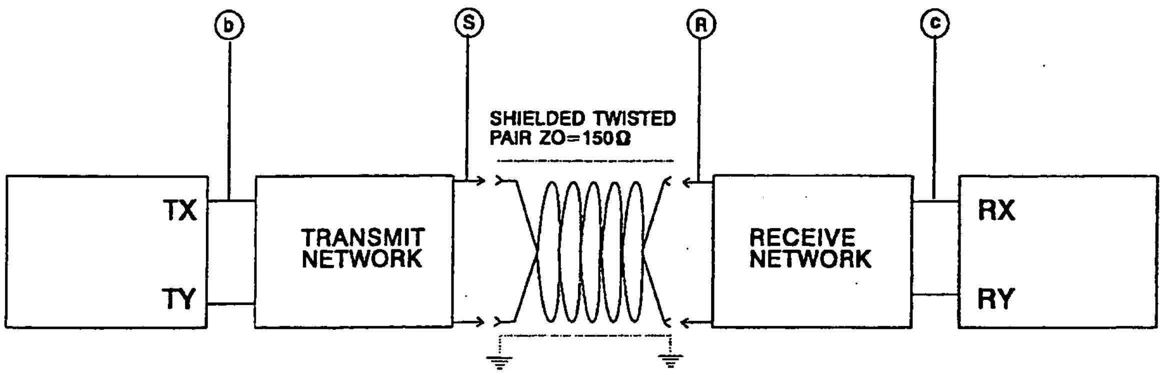


Figure 30 - The STP link

These two connector types (BNC and TNC) are specified to provide polarization to prevent the incorrect connection or transmitter-to-transmitter or receiver-to-receiver. The transmitting end of such a cable shall be implemented with a male BNC type connector and the receiving end shall be implemented with a male TNC type connector. The transmitter and receiver shall be implemented using female BNC and TNC type connectors respectively.

7.2.4.2 Connectors for miniature style coaxial cable

The miniature style coaxial cable data links are connected using a 75Ω miniature coax connector with an optional shield (see figure 27) that interfaces with industry standard headers with 0,64 mm (0,025 in) square posts on 2,54 mm (0,100 in) spacing. Primary application of the miniature coaxial cable is intended be for interconnection within a cabinet.

Due to cost reasons, these connectors are not entirely shielded and leakage of RFI will occur. A shielded cabinet and leakage control techniques such as ferrite beads or lossy tubing is recommended for compliance with EMC standards even with double shielded miniature coax (refer to annex F.4).

7.3 150Ω shielded twisted pair data link

This data link is based on the existing "Type 1" and "Type 2" 150Ω STP cable as defined in EIA/TIA 568. This data link is intended for use only at the lower data rates of 133 Mbaud and 266 Mbaud. One cable contains two twisted pairs, one pair makes the transmitting medium. The other pair makes the receiving medium.

The twisted pair shall be connected to the transmitter differential outputs and to the receiver's differential inputs inputs. The cable shield shall be grounded at both ends (see figure 30).

7.3.1 ECL compatible driver characteristics

The driver characteristics are based on ECL output specifications. The driver shall have complementary ECL outputs. The output signal is characterized and tested as in figure 31. The STP link shall comply with the normalized masks in table 11 and figure 29 at 133 Mbaud and 266 Mbaud.

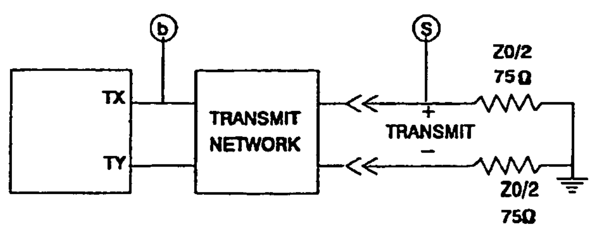


Figure 31 - STP cable transmitter test p ints

7.3.2 ECL compatible receiver characteristics

At the receiver end the cable is terminated by an equivalent impedance of 150Ω. An equalizer network may be used in order to compensate for cable response and to gain distance by improving the S/N ratio at the data recovery circuit input. The input circuit is ECL compatible and should be compatible with the specifications in table 10.

7.3.3 STP connector

The connector for STP is the 9-pin shielded "D" connector conforming to MIL-C-24308. The plug (male) half of the connector shall be mounted on the cable. One connector is required to connect both transmitting and receiving shielded twisted pairs at one node.

The connector pin assignments are shown in figure 32.

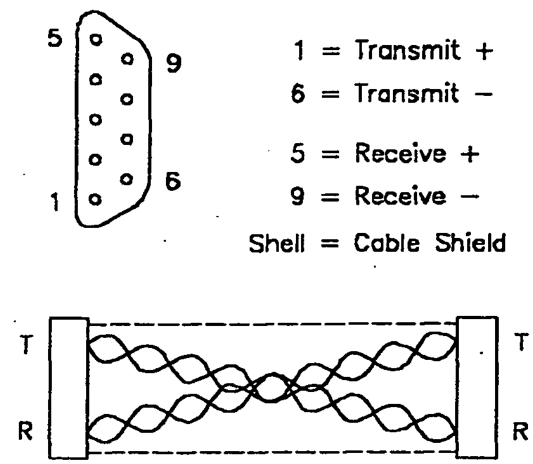


Figure 32 - Connector for shielded twisted pair

8 Optical fibre cable plant sp cification

8.1 SM cable plant specification

This sub-clause specifies a single-mode cable plant (see 3.1.17 for definition) for the Fibre Channel data rates of 266, 531 and 1 063 Mbaud at their rated distances of 2km and 10 km.

The cable plant is generally insensitive to data rate and therefore any installed portions of the cable plant may be used at any data rate (see table 12).

Table 12 - Single-mode cable plant

FC-0		100-SM-LL-L	100-SM-LL-I	50-SM-LL-L	25-SM-LL-L	25-SM-LL-I
Subclause		6.1	6.1	6.1	6.1	6.1
Operating Range	m	2-10k	2-2k	2-10k	2-10k	2-2k
Cable Plant Dispersion	ps/nm	35	12	60	60	12
Dispersion Related Penalty	dB	1	1	1	1	1
Reflection Related Penalty	dB	1	1	1	1	1
Loss Budget	dB	14	6	14	14	6

8.1.1 Optical fibre type

The optical fibre shall conform to EIA/TIA 492BAAA.

8.1.2 Cable plant loss budget

The loss budget for single-mode fibre shall be no greater than specified in table 12. These limits were arrived at by taking the difference between the minimum transmitter output power and the receiver sensitivity and subtracting 2dB to account for the dispersion and reflection link penalties. See annex E.3 for cable plant examples.

The loss of the cable plant shall be verified by the methods of OFSTP-7.

There are no requirements for fixed attenuators in the single-mode classes. All receivers and transmitters, of a given data rate, may intercommunicate without operability limit as long as the distance requirement of the shorter class is not exceeded.

8.1.3 Optical return loss

The cable plant optical return loss, with the receiver connected, shall be greater than or equal to 12dB. This is required to keep the reflection penalty under control. The receiver shall have a return loss greater than or equal to one glass air interface.

Connectors and splices shall each have a return loss greater than 26dB as measured by the methods of FOTP-107.

8.2 MM 62,5μm Cable plant specification

The 62,5μm Cable plant is the preferred cable plant for the 1 300nm LED based components. The short wavelength 780nm CD laser can also be used on the 62,5μm cable plant with some limited distance restrictions. See annex C and annex E.

Table 13 - 62,5μm multimode cable plant

FC-0	25-M6-LE-I	12-M6-LE-I
Subclause	6.3	6.3
Data Rate	25MB/sec	12MB/sec
Operating Range	2m-1,5km	2m-1,5km
Loss Budget	6dB	6dB

8.2.1 Optical fibre type

The optical fibre shall conform to EIA/TIA - 492AAAA.

Table 14 - 62,5μm fibre type		
Nominal Core Diameter FOTP-58	Cladding Diameter FOTP-45 & FOTP-176 or FOTP-48	Nominal Numerical Aperture FOTP-177
62,5μm	125μm	0,275

8.2.2 Modal Bandwidth

The following normalized bandwidth values are based on a nominal source wavelength of 850nm and 1 300nm (see table 15).

Table 15 - 62,5μm multimode bandwidth		
Wavelength	Modal bandwidth (-3dB optical min)	Test per
850nm	160MHz*km	FOTP-30 or FOTP-51 with FOTP-54
1 300nm	500MHz*km	FOTP-30 or FOTP-51 with FOTP-54

Note - Some users may wish to install higher modal bandwidth fibre to facilitate future use of the cable plant for higher bandwidth applications. For shorter distances, a lower bandwidth fibre may be substituted provided the performance requirements are met.

8.2.3 Cable plant loss budget

The loss budget for the multimode fibre cable plant shall be no greater than specified in table 13. These limits were arrived at by taking the difference between the minimum transmitter output power and the receiver sensitivity. The limits include the losses of the fiber and other components in the link such as splices and connectors. The connectors at the ends of the links are included in the transmitter and receiver specifications and not in the cable plant limit.

In some cases the modal dispersion limit may be reached in an installation before the installation loss limit of table 13. See annex E.4 for examples.

Conformance to the loss budget requirements shall be verified by means of OFSTP-14.

8.2.4 Optical return loss

The cable plant optical return loss, with the receiver connected, shall be greater than or equal to 12dB. This is required to keep the reflection penalty under control. The receiver shall have a return loss greater than or equal to one glass air interface.

Connectors and splices shall each have a return loss greater than 20dB.

8.2.5 Optical fibre chromatic dispersion parameters

The effects of chromatic dispersion on total system bandwidth shall be considered. For 1 300nm LED sources, this effect may be significant due to their relatively wide spectral width (see table 9 and figure 26). For 780nm lasers, this effect is less significant due to their very narrow spectral width (see table 9).

8.3 MM 50μm cable plant specification

The 50μm cable plant is the preferred cable plant for the 780nm short wavelength CD laser. The long wave length 1 300nm LED based components may also be used on the 50μm cable plant with some distance restrictions. See annex C and annex E.

Table 16 - 50μm multimode cable plant			
FC-0	100-M5-SL-I	50-M5-SL-I	25-M5-SL-I
Subclause	6.2	6.2	6.2
Data Rate	100MB/sec	50MB/sec	25MB/sec
Operating Range	2m - 500m	2m - 1km	2m - 2km
Loss Budget	6 dB	8 dB	12 dB

8.3.1 Optical fibr typ

The 50μm optical fibre shall meet the requirements in tables 16, 17, and 18.

Table 17 - 50μm fibre type		
Nominal Core Diameter FOTP-58	Cladding Diameter FOTP-45 & FOTP-176 or FOTP-48	Nominal Numerical Aperture FOTP-177
50μm	125μm	0,20

8.3.2 Modal Bandwidth

The following normalized bandwidth values are based on a nominal source wavelength of 850 and 1 300nm (see table 18).

Table 18 - 50μm multimode bandwidth		
Wavelength	Modal bandwidth (-3dB optical min)	Test per
850nm	500MHz*km	FOTP-30 or FOTP-51 with FOTP-54
1 300nm	500MHz*km	FOTP-30 or FOTP-51 with FOTP-54

NOTE - The bandwidth shown in the table meets the performance requirements for 780nm laser operation.

8.3.3 Cable plant loss budget

The loss budget for the 50μm multimode fibre cable plant shall be no greater than specified in table 16. These limits were arrived at by taking the difference between the minimum transmitter output power and the receiver sensitivity. This includes the losses of the fiber and other components in the link such as splices and connectors. The connectors at the ends of the links are included in the transmitter and receiver specifications and not in the cable plant limit.

In some cases the modal dispersion limit may be reached in an installation before the installation loss limit of table 16. See annex E.4 for examples.

The loss of the cable plant shall be verified by the methods of OFSTP-14.

8.3.4 Optical return loss

The cable plant optical return loss, with the receiver connected, shall be greater than or equal to 12dB. This is required to keep the reflection penalty under control. The receiver shall have a return loss greater than or equal to one glass air interface.

Connectors and splices shall each have a return loss greater than 20dB.

8.3.5 Optical fibre chromatic dispersion parameters

The effects of chromatic dispersion for 50μm multimode fibres are similar to 62.5μm multimode fibres. Therefore, refer to 8.2.5 for discussion on their effects.

8.4 Connectors and splices

Connectors and splices of any nature are allowed inside the cable plant as long as the resulting loss conforms to the optical budget of this standard. The number and quality of connections represent a design trade-off outside the scope of this document.

9 Electrical cable plant specification

This clause defines the link requirements for a Fibre Channel electrical cable plant.

9.1 Video Cable Plant Specification

This subclause specifies a video style coaxial cable plant which is capable of communication at distances and data rates specified in Table 19. The cable plant is generally insensitive to data rate and therefore any installed portions of the cable plant may be used at any data rate

Table 19 - Video style coaxial cable plant

FC-0		100-TV-EL-S	50-TV-EL-S	25-TV-EL-S	12-TV-EL-S
Subclause		7.2	7.2	7.2	7.2
Distance	m	0-25	0-50	0-75	0-100
Attenuation @ half baud rate	dB/m	0,288	0,167	0,096	0,088
Connector Related Loss	dB/conn	0,25	0,25	0,25	0,25

9.1.1 Video Coax Type

The coax shall conform to RG 6/U type or RG 59/U type coaxial cable specifications.

9.1.2 Discrete Connector Return Loss

The minimum return loss from connectors with the receivers connected shall be greater than 20 dB. This is required to keep the reflections from distorting the transmitted signal.

9.2 Miniature Coax Cable Plant Specification

This subclause specifies a miniature coaxial cable plant which is capable of communication at distances and data rates as specified in Table 20 . The cable plant is generally insensitive to data rate and therefore any installed portions of the cable plant may be used at any data rate

Table 20 - Miniature style coaxial cable plant

FC-0		100-MI-EL-S	50-MI-EL-S	25-MI-EL-S	12-MI-EL-S
Subclause		7.2	7.2	7.2	7.2
Distance	m	0-10	0-15	0-25	0-35
Attenuation @ half baud rate	dB/m	0,62	0,46	0,31	0,21
Connector Related Loss	dB/conn	0,5	0,5	0,25	0,25

9.2.1 Miniature Coax Type

The coax shall conform to RG 179B/U type coaxial cable specifications.

9.2.2 Discrete Connector Return Loss

The minimum return loss from connectors with the receivers connected shall be greater than 20 dB. This is required to keep the reflections from distorting the transmitted signal.

9.3 Video and Miniature Coax Interoperability

A specific goal of the coaxial cable plant is to allow interoperability with restricted lengths of miniature coax cable and video style coax. For example, as long as the miniature coax cable is less than 2 m in length at each end of two cabinets, the full 50 m capability of the video style coax should be achievable for cabinet interconnections.

Especially at transmission rates of 531 Mbaud or greater, particular attention must be given to the transition between miniature coax and video style coax. An example of the greater care is the use of microstrip lines of proper impedance. No more than four miniature coaxial connectors should be present from the ECL transmitter to the receiver of the data link.

Careful attention to details as noted above should allow the system designer to construct a low-cost and yet high performance system capable of reliable error rate performance. The coaxial data link will meet the requirements of many shorter distance Fibre Channel applications.

9.4 Shielded Twisted Pair Cable Plant Specification

This subclause specifies an STP cable plant which is capable of communication at frequency dependent distances as shown in table 21.

Table 21 - STP style cable plant

FC-0		25-TP-EL-S	12-TP-EL-S
Subclause		7.3	7.3
Distance	m	0-50	0-100
Attenuation @ half baud rate	dB/m	0,138	0,092
Connector Related Loss	dB/ conn.	0,25	0,15

The STP cable shall conform to EIA/TIA 568 (see clause 2).

10 Optical Interfac Conn ctor Specification

The primary function of the optical interface connector is to align the optical transmission fibre mechanically to an optical port on a component such as a receiver or a transmitter.

Note in this clause and figures only unique Fibre Channel Duplex dimensions are provided. All others are in the referenced JIS-C-5973 standard.

The objective of this clause is to specify the connector and interfaces sufficiently to insure the following:

- Both Mechanical and Optical Performance
- Intermatability
- To allow the maximum supplier flexibility.

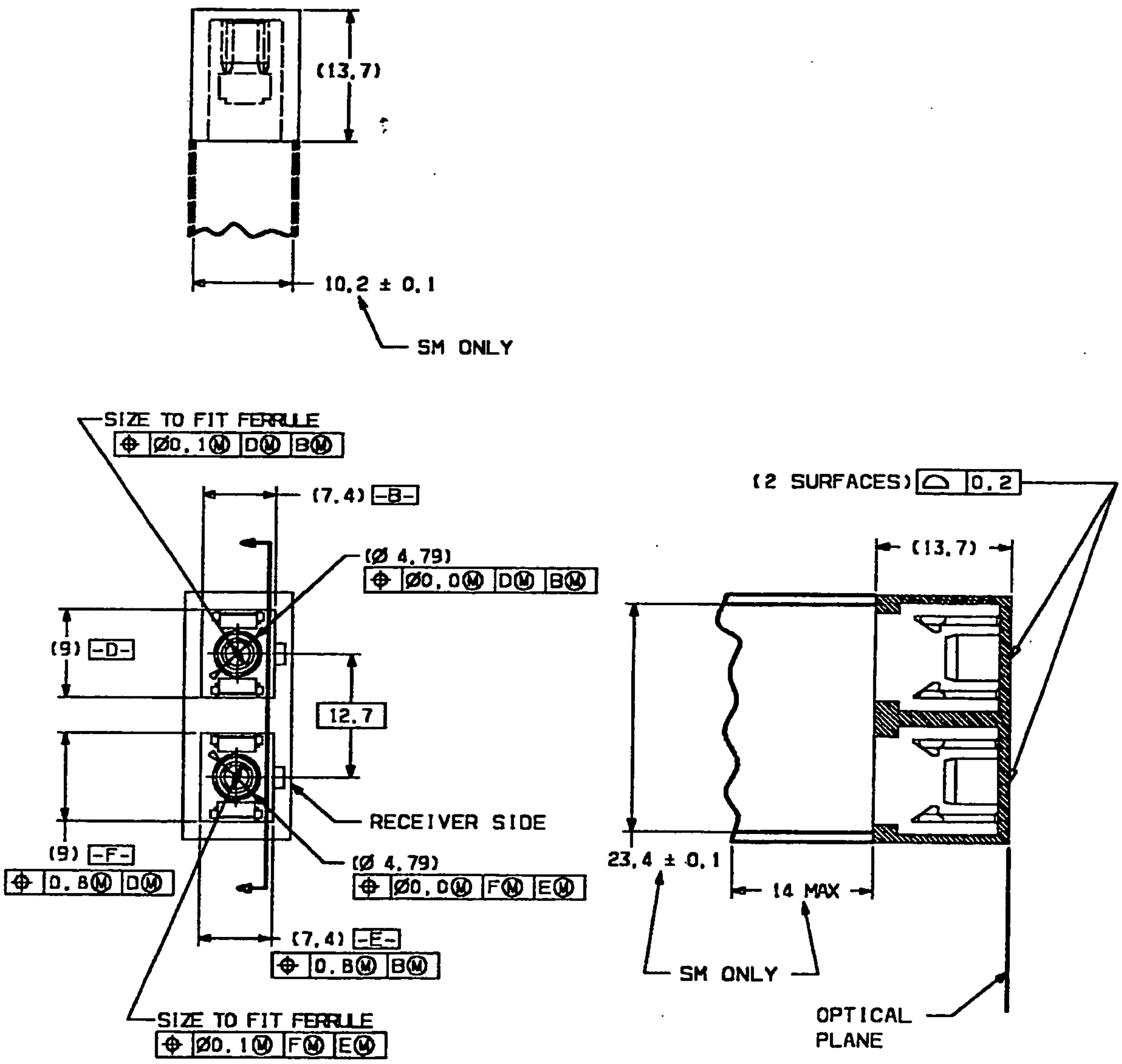


Figure 33 - Dupl x Receptacle

10.1.1 Relationship t other standard connect rs

The optical interface connector shall dimensionally conform to the industry standard type SC connector documented in JIS-C-5973, as it relates to intermatability, unless otherwise specified.

10.1.2 Testing Recommendations

Supporting test information is contained in annex G.

10.2 Optical receptacle

Figure 33 dimensionally specifies the receptacle of the optical interface connector.

Single mode keying
As a safety feature for laser operation the optical interface contains keying features to prevent complete insertion of a multimode plug into a single mode receptacle. This is accomplished by narrowing the key protrusion on the plug and the key notch. The receptacle keying dimensions are specified in figure 34. The dimensions for the single mode key shall take pr cedence over the key dimensions of the standard referenced in clause 10.1.1.

10.3 Optical Plug

Figure 35 on page 61 dimensionally specifies the male plug of the optical interface connector.

The two simplex plugs may be connected in a resilient manner to allow relative movement. However it is recognized that a possible means of accomplishing this will include the attachment of a holding or clamping mechanism between two simplex plugs. Such arrangements will of necessity increase the physical size of the individual plugs above the attachment point and present the possibility of mechanical interference with objects in the vicinity of the receptacle. An example of such an object would be a cover panel where it is desired to keep the cutout as small as possible to minimize radiation leakage.

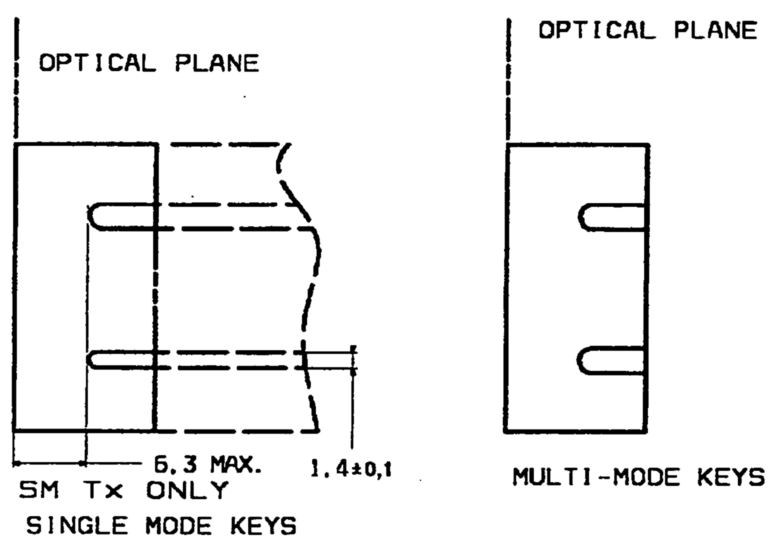


Figure 34 - Receptacle Single Mode Keying Requirements

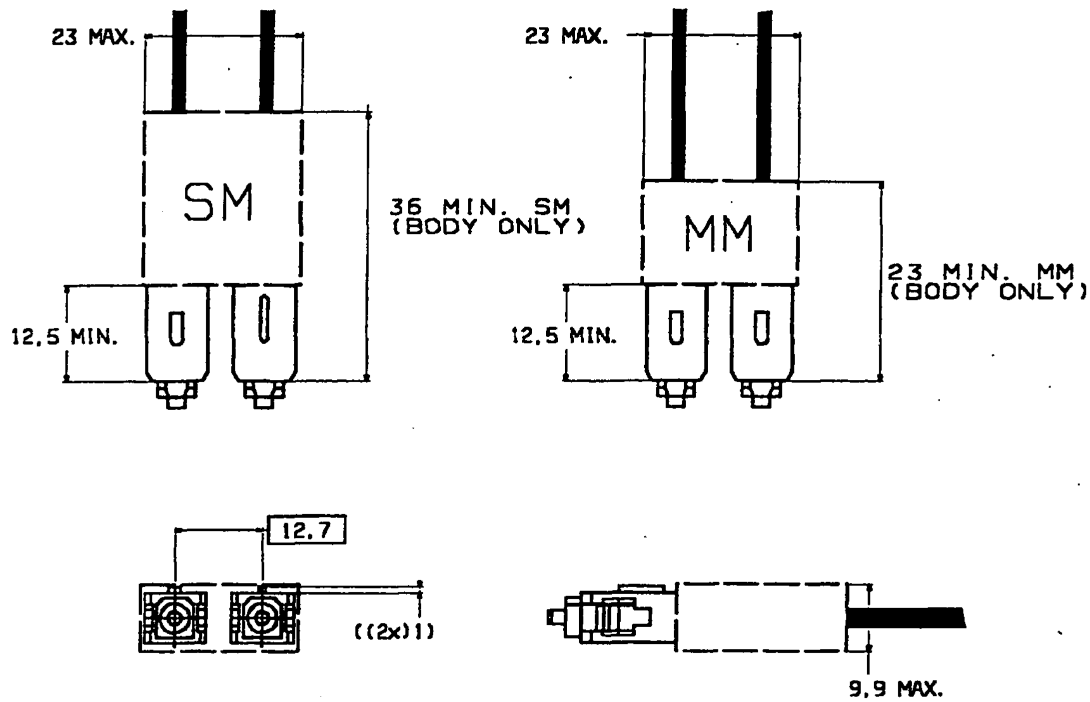


Figure 35 - Optical Plug Dimensional Requirements

In order to provide for this situation the plug shall not exceed the dimensions of the simplex plug for a distance of 12,5mm (measured from the end of the connector housing) and the interface connector plug shall fit through an opening of 23,0mm by 9,9mm located greater than 12,5 mm from the mechanical seating plane.

Warning: In the design of a duplex plug assembly, great care must be taken to ensure pluggability with the receptacle. The ferrule to its grip body float to be a minimum of 0,2mm (0,1mm radial) for each connector, and one connector grip body to another connector grip body float to be a minimum of 0,9mm (0,45mm radial) for each connector. These floats are required for interoperability.

10.3.1 Ferrule

The ferrule diameter dimensions are specified in figure 36 for both the multimode and single mode ferrules.

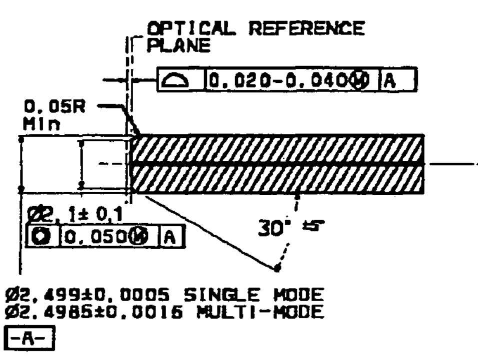


Figure 36 - Ferrule Dimensional Requirements

The ferrule end shall seat to the optical reference plane with a static force of 6,7N minimum to 13,1N maximum per ferrule.

10.3.2 Single m de keying

As a safety feature for laser operation the optical interface connector contains keying features to prevent inadvertent insertion of a multimode plug into a single mode receptacle. This is accomplished by narrowing the key protrusion on the plug and the key notch. The plug keying dimensions are specified in figure 37. The dimensions for the single mode key shall take precedence over the key dimensions of the standard referenced in 10.1.1.

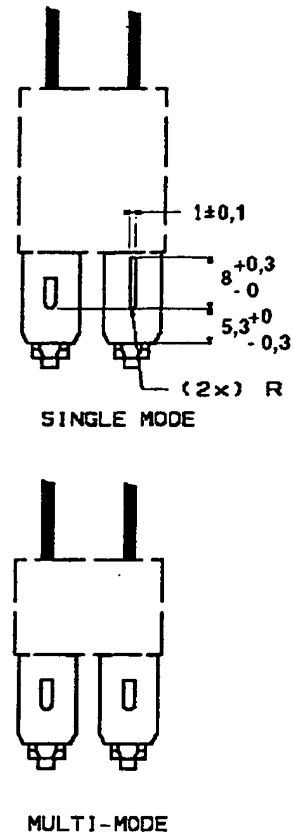


Figure 37 - Plug Single Mode Keying Requirements

11 FC-1 8B/10B Transmission Code

Information to be transmitted over a Fibre shall be encoded eight bits at a time into a 10-bit Transmission Character and then sent serially by bit. Information received over a Fibre shall be collected ten bits at a time, and those Transmission Characters that are used for data, called Data Characters, shall be decoded into the correct eight-bit codes. The 10-bit Transmission Code supports all 256 eight-bit combinations. Some of the remaining Transmission Characters, referred to as Special Characters, are used for functions which are to be distinguishable from the contents of a frame.

NOTES

1 The following definition of the 10-bit Transmission Code is based on (and is in basic agreement with) Widmer, 1983 [1] and Franasak, 1984 [2]. Both of these references describe the same 10-bit transmission code.

2 The primary rationale for use of a Transmission Code is to improve the transmission characteristics of information to be transferred across a Fibre. The encodings defined by the Transmission Code ensure that sufficient transitions are present in the serial bit stream to make clock recovery possible at the receiver. Such encoding also greatly increases the likelihood of detecting any single or multiple bit errors that may occur during transmission and reception of information. In addition, some of the Special Characters of the Transmission Code contain a distinct and easily recognizable bit pattern (a Comma) which assists a receiver in achieving word alignment on the incoming bit stream.

11.1 Notation conventions

FC-1 uses letter notation for describing information bits and control variables. Such notation differs from the bit notation specified by the remainder of this standard (see 3.2). The following text describes the translation process between these notations and provides a translation example. It also describes the conventions used to name valid Transmission Characters. This text is provided for the purposes of terminology clarification only and is not intended to restrict the implementation of FC-1 functions in any way.

An unencoded FC-1 information byte is composed of eight information bits A,B,C,D,E,F,G,H and the control variable Z. This information is encoded by FC-1 into the bits a,b,c,d,e,i,f,g,h,j of a 10-bit Transmission Character.

An information bit contains either a binary zero or a binary one. A control variable has either the value D or the value K. When the control variable associated with an unencoded FC-1 information byte contains the value D, that byte is referred to as a Valid Data Byte. When the control variable associated with an unencoded FC-1 information byte contains the value K, that byte is referred to as a Special Code.

The information bit labeled A corresponds to bit 0 in the numbering scheme of the FC-2 specification, B corresponds to bit 1, and so on, as shown in figure 38. The control variable is typically not specified by FC-2. When the control variable is not specified by FC-2, FC-1 assumes its value to be D (data).

```
FC-2 bit                          control
notation:        7 6 5 4 3 2 1 0  variable

FC-1 unencoded
bit notation:    H G F E D C B A     Z
```

Figure 38 - Bit designations

Each valid Transmission Character has been given a name using the following convention: Zxx.y, where Z is the control variable of the unencoded FC-1 information byte, xx is the decimal value of the binary number composed of the bits E, D, C, B, and A of the unencoded FC-1 information byte in that order, and y is the decimal value of the binary number composed of the bits H, G, and F of the unencoded FC-1 information byte in that order. The value of Z is used to indicate whether the Transmission Character is a Data Character (Z = D) or a Special Character (Z = K).

Figur 39 shows the conversion from FC-2 byte notation to the FC-1 Transmission Character naming convention described above.

```
FC-2 byte notation:              hex 'BC' -- Special
                                               Code

FC-2 bit notation:               7654 3210 control
                                 ---- ---- -------
                                 1011 1100    K

FC-1 unencoded                   HGF EDCBA    Z
bit notation:                    --- -----    -
                                 101 11100    K

FC-1 unencoded bit notation      Z    EDCBA HGF
reordered to conform with        -    ----- ---
Zxx.y naming convention:         K    11100 101

FC-1 Transmission
Character name:                  K      28  .5
```

Figure 39 - Conversion example

NOTE - Most Kxx.y combinations do not result in valid Transmission Characters within the 8B/10B Transmission Code. Only those combinations which result in Special Characters as specified by table 23 are considered valid.

11.2 Character encoding and decoding

The following information describes how the tables shall be used for both generating valid Transmission Characters (encoding) and checking the validity of received Transmission Characters (decoding). It also specifies the ordering rules to be followed when transmitting the bits within a character and the characters within the higher-level constructs specified by the document (i.e., Ordered Sets and frames).

11.2.1 Transmission order

Within the definition of the 8B/10B Transmission Code, the bit positions of the Transmission Characters are labeled a,b,c,d,e,i,f,g,h, and j. Bit "a" shall be transmitted first, followed by bits "b," "c," "d," "e," "i," "f," "g," "h," and "j," in that order (note that bit "i" shall be transmitted between bit "e" and bit "f," rather than in the order that would be indicated by the letters of the alphabet).

Characters within Ordered Sets (as specified by 11.4) shall be transmitted sequentially beginning with the Special Character used to distinguish the Ordered Set (e.g., K28.5) and proceeding character by character from left to right within the definition of the Ordered Set until all characters of the Ordered Set are transmitted.

The contents of a frame (as specified in clause 17) shall be transmitted sequentially beginning with the Ordered Set used to denote the start of frame (the SOF delimiter) and proceeding character by character from left to right within the definition of the frame until the Ordered Set used to denote the end of frame (the EOF delimiter) is transmitted.

11.2.2 Valid and invalid Transmission Characters

Tables 22 and 23 define the valid Data Characters and valid Special Characters (K characters), respectively. These tables shall be used for both generating valid Transmission Characters (encoding) and checking the validity of received Transmission Characters (decoding). In the tables, each Valid-Data-Byte or Special-Code entry has two columns that represent two (not necessarily different) Transmission Characters. The two columns correspond to the current value of the Running Disparity ("Current RD −" or "Current RD +"). Running Disparity is a binary parameter with either the value negative (−) or the value positive (+).

After powering on or exiting diagnostic mode, the transmitter shall assume the negative value for its initial Running Disparity. Upon transmission of any Transmission Character, the transmitter shall calculate a new value for its Running Disparity based on the contents of the transmitted character.

After powering on or exiting diagnostic mode, the receiver should assume either the positive or negative value for its initial Running Disparity. Upon reception of any Transmission Character, the receiver shall determine whether the Transmission Character is valid or invalid according to the following rules and tables and shall calculate a new value for its Running Disparity based on the contents of the received character.

The following rules for Running Disparity shall be used to calculate the n w Running Disparity

value for Transmission Characters that have been transmitted (transmitter's Running Disparity) and that have been received (receiver's Running Disparity).

Running Disparity for a Transmission Character shall be calculated on the basis of sub-blocks, where the first six bits (abcdei) form one sub-block (six-bit sub-block) and the second four bits (fghj) form the other sub-block (four-bit sub-block). Running Disparity at the beginning of the six-bit sub-block is the Running Disparity at the end of the last Transmission Character. Running Disparity at the beginning of the four-bit sub-block is the Running Disparity at the end of the six-bit sub-block. Running Disparity at the end of the Transmission Character is the Running Disparity at the end of the four-bit sub-block.

Running Disparity for the sub-blocks shall be calculated as follows:

— Running Disparity at the end of any sub-block is positive if the sub-block contains more ones than zeros. It is also positive at the end of the six-bit sub-block if the six-bit sub-block is 000111, and it is positive at the end of the four-bit sub-block if the four-bit sub-block is 0011.

— Running Disparity at the end of any sub-block is negative if the sub-block contains more zeros than ones. It is also negative at the end of the six-bit sub-block if the six-bit sub-block is 111000, and it is negative at the end of the four-bit sub-block if the four-bit sub-block is 1100.

— Otherwise, Running Disparity at the end of the sub-block is the same as at the beginning of the sub-block.

NOTE - All sub-blocks with equal numbers of zeros and ones are disparity neutral. In order to limit the run length of 0's or 1's between sub-blocks, the 8B/10B transmission code rules specify that sub-blocks encoded as 000111 or 0011 are generated only when the Running Disparity at the beginning of the sub-block is positive; thus, Running Disparity at the end of these sub-blocks will also be positive. Likewise, sub-blocks containing 111000 or 1100 are generated only when the Running Disparity at the beginning of the sub-block is negative; thus, Running Disparity at the end of these sub-blocks will also be negative.

11.2.2.1 Use of the tables for generating Transmission Characters

The appropriate entry in the table shall be found for the Valid Data Byte or Special Code for which a Transmission Character is to be generated (encoded). The current value of the transmitter's Running Disparity shall be used to select the Transmission Character from its corresponding column. For each Transmission Character transmitted, a new value of the Running Disparity shall be calculated. This new value shall be used as the transmitter's Current Running Disparity for the next Valid Data Byte or Special Code to be encoded and transmitted.

11.2.2.2 Use of the tables for checking

the validity of received Transmission Characters

The column corresponding to the current value of the receiver's Running Disparity shall be searched for the received Transmission Character. If the received Transmission Character is found in the proper column, then the Transmission Character shall be considered valid and the associated data byte or Special Code determined (decoded). If the received Transmission Character is not found in that column, then the Transmission Character shall be considered invalid and a Code Violation detected and reported to its associated port. Independent of the Transmission Character's validity, the received Transmission Character shall be used to calculate a new value of Running Disparity. This new value shall be used as the receiver's Current Running Disparity for the next received Transmission Character.

NOTE - Detection of a Code Violation does not necessarily indicate that the Transmission Character in which the Code Violation was detected is in error. Code Violations may result from a prior error which altered the Running Disparity of the bit stream but which did not result in a detectable error at the Transmission Character in which the error occurred. The example shown in figure 40 exhibits this behavior:

	RD	Character	RD	Character	RD	Character	RD
Transmitted character stream	-	D21.1	-	D10.2	-	D23.5	+
Transmitted bit stream	-	101010 1001	-	010101 0101	-	111010 1010	+
Bit stream after error	-	101010 1011	+	010101 0101	+	111010 1010	+
Decoded character stream	-	D21.0	+	D10.2	+	Code Violation	+

Figure 40 - Delayed Code Violation example

Table 22 (Page 1 of 3) - Valid Data Characters

Data Byte Name	Bits HGF EDCBA	Current RD - abcdei fghj	Current RD + abcdei fghj	Data Byte Name	Bits HGF EDCBA	Current RD - abcdei fghj	Current RD + abcdei fghj
D0.0	000 00000	100111 0100	011000 1011	D16.1	001 10000	011011 1001	100100 1001
D1.0	000 00001	011101 0100	100010 1011	D17.1	001 10001	100011 1001	100011 1001
D2.0	000 00010	101101 0100	010010 1011	D18.1	001 10010	010011 1001	010011 1001
D3.0	000 00011	110001 1011	110001 0100	D19.1	001 10011	110010 1001	110010 1001
D4.0	000 00100	110101 0100	001010 1011	D20.1	001 10100	001011 1001	001011 1001
D5.0	000 00101	101001 1011	101001 0100	D21.1	001 10101	101010 1001	101010 1001
D6.0	000 00110	011001 1011	011001 0100	D22.1	001 10110	011010 1001	011010 1001
D7.0	000 00111	111000 1011	000111 0100	D23.1	001 10111	111010 1001	000101 1001
D8.0	000 01000	111001 0100	000110 1011	D24.1	001 11000	110011 1001	001100 1001
D9.0	000 01001	100101 1011	100101 0100	D25.1	001 11001	100110 1001	100110 1001
D10.0	000 01010	010101 1011	010101 0100	D26.1	001 11010	010110 1001	010110 1001
D11.0	000 01011	110100 1011	110100 0100	D27.1	001 11011	110110 1001	001001 1001
D12.0	000 01100	001101 1011	001101 0100	D28.1	001 11100	001110 1001	001110 1001
D13.0	000 01101	101100 1011	101100 0100	D29.1	001 11101	101110 1001	010001 1001
D14.0	000 01110	011100 1011	011100 0100	D30.1	001 11110	011110 1001	100001 1001
D15.0	000 01111	010111 0100	101000 1011	D31.1	001 11111	101011 1001	010100 1001
D16.0	000 10000	011011 0100	100100 1011	D0.2	010 00000	100111 0101	011000 0101
D17.0	000 10001	100011 1011	100011 0100	D1.2	010 00001	011101 0101	100010 0101
D18.0	000 10010	010011 1011	010011 0100	D2.2	010 00010	101101 0101	010010 0101
D19.0	000 10011	110010 1011	110010 0100	D3.2	010 00011	110001 0101	110001 0101
D20.0	000 10100	001011 1011	001011 0100	D4.2	010 00100	110101 0101	001010 0101
D21.0	000 10101	101010 1011	101010 0100	D5.2	010 00101	101001 0101	101001 0101
D22.0	000 10110	011010 1011	011010 0100	D6.2	010 00110	011001 0101	011001 0101
D23.0	000 10111	111010 0100	000101 1011	D7.2	010 00111	111000 0101	000111 0101
D24.0	000 11000	110011 0100	001100 1011	D8.2	010 01000	111001 0101	000110 0101
D25.0	000 11001	100110 1011	100110 0100	D9.2	010 01001	100101 0101	100101 0101
D26.0	000 11010	010110 1011	010110 0100	D10.2	010 01010	010101 0101	010101 0101
D27.0	000 11011	110110 0100	001001 1011	D11.2	010 01011	110100 0101	110100 0101
D28.0	000 11100	001110 1011	001110 0100	D12.2	010 01100	001101 0101	001101 0101
D29.0	000 11101	101110 0100	010001 1011	D13.2	010 01101	101100 0101	101100 0101
D30.0	000 11110	011110 0100	100001 1011	D14.2	010 01110	011100 0101	011100 0101
D31.0	000 11111	101011 0100	010100 1011	D15.2	010 01111	010111 0101	101000 0101
D0.1	001 00000	100111 1001	011000 1001	D16.2	010 10000	011011 0101	100100 0101
D1.1	001 00001	011101 1001	100010 1001	D17.2	010 10001	100011 0101	100011 0101
D2.1	001 00010	101101 1001	010010 1001	D18.2	010 10010	010011 0101	010011 0101
D3.1	001 00011	110001 1001	110001 1001	D19.2	010 10011	110010 0101	110010 0101
D4.1	001 00100	110101 1001	001010 1001	D20.2	010 10100	001011 0101	001011 0101
D5.1	001 00101	101001 1001	101001 1001	D21.2	010 10101	101010 0101	101010 0101
D6.1	001 00110	011001 1001	011001 1001	D22.2	010 10110	011010 0101	011010 0101
D7.1	001 00111	111000 1001	000111 1001	D23.2	010 10111	111010 0101	000101 0101
D8.1	001 01000	111001 1001	000110 1001	D24.2	010 11000	110011 0101	001100 0101
D9.1	001 01001	100101 1001	100101 1001	D25.2	010 11001	100110 0101	100110 0101
D10.1	001 01010	010101 1001	010101 1001	D26.2	010 11010	010110 0101	010110 0101
D11.1	001 01011	110100 1001	110100 1001	D27.2	010 11011	110110 0101	001001 0101
D12.1	001 01100	001101 1001	001101 1001	D28.2	010 11100	001110 0101	001110 0101
D13.1	001 01101	101100 1001	101100 1001	D29.2	010 11101	101110 0101	010001 0101
D14.1	001 01110	011100 1001	011100 1001	D30.2	010 11110	011110 0101	100001 0101
D15.1	001 01111	010111 1001	101000 1001	D31.2	010 11111	101011 0101	010100 0101

99

Table 22 (Page 2 of 3) - Valid Data Characters							
Data Byte Name	Bits HGF EDCBA	Current RD - abcdei fghj	Current RD + abcdei fghj	Data Byte Name	Bits HGF EDCBA	Current RD - abcdei fghj	Current RD + abcdei fghj
D0.3	011 00000	100111 0011	011000 1100	D16.4	100 10000	011011 0010	100100 1101
D1.3	011 00001	011101 0011	100010 1100	D17.4	100 10001	100011 1101	100011 0010
D2.3	011 00010	101101 0011	010010 1100	D18.4	100 10010	010011 1101	010011 0010
D3.3	011 00011	110001 1100	110001 0011	D19.4	100 10011	110010 1101	110010 0010
D4.3	011 00100	110101 0011	001010 1100	D20.4	100 10100	001011 1101	001011 0010
D5.3	011 00101	101001 1100	101001 0011	D21.4	100 10101	101010 1101	101010 0010
D6.3	011 00110	011001 1100	011001 0011	D22.4	100 10110	011010 1101	011010 0010
D7.3	011 00111	111000 1100	000111 0011	D23.4	100 10111	111010 0010	000101 1101
D8.3	011 01000	111001 0011	000110 1100	D24.4	100 11000	110011 0010	001100 1101
D9.3	011 01001	100101 1100	100101 0011	D25.4	100 11001	100110 1101	100110 0010
D10.3	011 01010	010101 1100	010101 0011	D26.4	100 11010	010110 1101	010110 0010
D11.3	011 01011	110100 1100	110100 0011	D27.4	100 11011	110110 0010	001001 1101
D12.3	011 01100	001101 1100	001101 0011	D28.4	100 11100	001110 1101	001110 0010
D13.3	011 01101	101100 1100	101100 0011	D29.4	100 11101	101110 0010	010001 1101
D14.3	011 01110	011100 1100	011100 0011	D30.4	100 11110	011110 0010	100001 1101
D15.3	011 01111	010111 0011	101000 1100	D31.4	100 11111	101011 0010	010100 1101
D16.3	011 10000	011011 0011	100100 1100	D0.5	101 00000	100111 1010	011000 1010
D17.3	011 10001	100011 1100	100011 0011	D1.5	101 00001	011101 1010	100010 1010
D18.3	011 10010	010011 1100	010011 0011	D2.5	101 00010	101101 1010	010010 1010
D19.3	011 10011	110010 1100	110010 0011	D3.5	101 00011	110001 1010	110001 1010
D20.3	011 10100	001011 1100	001011 0011	D4.5	101 00100	110101 1010	001010 1010
D21.3	011 10101	101010 1100	101010 0011	D5.5	101 00101	101001 1010	101001 1010
D22.3	011 10110	011010 1100	011010 0011	D6.5	101 00110	011001 1010	011001 1010
D23.3	011 10111	111010 0011	000101 1100	D7.5	101 00111	111000 1010	000111 1010
D24.3	011 11000	110011 0011	001100 1100	D8.5	101 01000	111001 1010	000110 1010
D25.3	011 11001	100110 1100	100110 0011	D9.5	101 01001	100101 1010	100101 1010
D26.3	011 11010	010110 1100	010110 0011	D10.5	101 01010	010101 1010	010101 1010
D27.3	011 11011	110110 0011	001001 1100	D11.5	101 01011	110100 1010	110100 1010
D28.3	011 11100	001110 1100	001110 0011	D12.5	101 01100	001101 1010	001101 1010
D29.3	011 11101	101110 0011	010001 1100	D13.5	101 01101	101100 1010	101100 1010
D30.3	011 11110	011110 0011	100001 1100	D14.5	101 01110	011100 1010	011100 1010
D31.3	011 11111	101011 0011	010100 1100	D15.5	101 01111	010111 1010	101000 1010
D0.4	100 00000	100111 0010	011000 1101	D16.5	101 10000	011011 1010	100100 1010
D1.4	100 00001	011101 0010	100010 1101	D17.5	101 10001	100011 1010	100011 1010
D2.4	100 00010	101101 0010	010010 1101	D18.5	101 10010	010011 1010	010011 1010
D3.4	100 00011	110001 1101	110001 0010	D19.5	101 10011	110010 1010	110010 1010
D4.4	100 00100	110101 0010	001010 1101	D20.5	101 10100	001011 1010	001011 1010
D5.4	100 00101	101001 1101	101001 0010	D21.5	101 10101	101010 1010	101010 1010
D6.4	100 00110	011001 1101	011001 0010	D22.5	101 10110	011010 1010	011010 1010
D7.4	100 00111	111000 1101	000111 0010	D23.5	101 10111	111010 1010	000101 1010
D8.4	100 01000	111001 0010	000110 1101	D24.5	101 11000	110011 1010	001100 1010
D9.4	100 01001	100101 1101	100101 0010	D25.5	101 11001	100110 1010	100110 1010
D10.4	100 01010	010101 1101	010101 0010	D26.5	101 11010	010110 1010	010110 1010
D11.4	100 01011	110100 1101	110100 0010	D27.5	101 11011	110110 1010	001001 1010
D12.4	100 01100	001101 1101	001101 0010	D28.5	101 11100	001110 1010	001110 1010
D13.4	100 01101	101100 1101	101100 0010	D29.5	101 11101	101110 1010	010001 1010
D14.4	100 01110	011100 1101	011100 0010	D30.5	101 11110	011110 1010	100001 1010
D15.4	100 01111	010111 0010	101000 1101	D31.5	101 11111	101011 1010	010100 1010

Table 22 (Pag 3 of 3) - Valid Data Characters							
Data Byte Name	Bits HGF EDCBA	Current RD - abcdei fghj	Current RD + abcdei fghj	Data Byte Name	Bits HGF EDCBA	Current RD - abcdei fghj	Current RD + abcdei fghj
D0.6	110 00000	100111 0110	011000 0110	D0.7	111 00000	100111 0001	011000 1110
D1.6	110 00001	011101 0110	100010 0110	D1.7	111 00001	011101 0001	100010 1110
D2.6	110 00010	101101 0110	010010 0110	D2.7	111 00010	101101 0001	010010 1110
D3.6	110 00011	110001 0110	110001 0110	D3.7	111 00011	110001 1110	110001 0001
D4.6	110 00100	110101 0110	001010 0110	D4.7	111 00100	110101 0001	001010 1110
D5.6	110 00101	101001 0110	101001 0110	D5.7	111 00101	101001 1110	101001 0001
D6.6	110 00110	011001 0110	011001 0110	D6.7	111 00110	011001 1110	011001 0001
D7.6	110 00111	111000 0110	000111 0110	D7.7	111 00111	111000 1110	000111 0001
D8.6	110 01000	111001 0110	000110 0110	D8.7	111 01000	111001 0001	000110 1110
D9.6	110 01001	100101 0110	100101 0110	D9.7	111 01001	100101 1110	100101 0001
D10.6	110 01010	010101 0110	010101 0110	D10.7	111 01010	010101 1110	010101 0001
D11.6	110 01011	110100 0110	110100 0110	D11.7	111 01011	110100 1110	110100 1000
D12.6	110 01100	001101 0110	001101 0110	D12.7	111 01100	001101 1110	001101 0001
D13.6	110 01101	101100 0110	101100 0110	D13.7	111 01101	101100 1110	101100 1000
D14.6	110 01110	011100 0110	011100 0110	D14.7	111 01110	011100 1110	011100 1000
D15.6	110 01111	010111 0110	101000 0110	D15.7	111 01111	010111 0001	101000 1110
D16.6	110 10000	011011 0110	100100 0110	D16.7	111 10000	011011 0001	100100 1110
D17.6	110 10001	100011 0110	100011 0110	D17.7	111 10001	100011 0111	100011 0001
D18.6	110 10010	010011 0110	010011 0110	D18.7	111 10010	010011 0111	010011 0001
D19.6	110 10011	110010 0110	110010 0110	D19.7	111 10011	110010 1110	110010 0001
D20.6	110 10100	001011 0110	001011 0110	D20.7	111 10100	001011 0111	001011 0001
D21.6	110 10101	101010 0110	101010 0110	D21.7	111 10101	101010 1110	101010 0001
D22.6	110 10110	011010 0110	011010 0110	D22.7	111 10110	011010 1110	011010 0001
D23.6	110 10111	111010 0110	000101 0110	D23.7	111 10111	111010 0001	000101 1110
D24.6	110 11000	110011 0110	001100 0110	D24.7	111 11000	110011 0001	001100 1110
D25.6	110 11001	100110 0110	100110 0110	D25.7	111 11001	100110 1110	100110 0001
D26.6	110 11010	010110 0110	010110 0110	D26.7	111 11010	010110 1110	010110 0001
D27.6	110 11011	110110 0110	001001 0110	D27.7	111 11011	110110 0001	001001 1110
D28.6	110 11100	001110 0110	001110 0110	D28.7	111 11100	001110 1110	001110 0001
D29.6	110 11101	101110 0110	010001 0110	D29.7	111 11101	101110 0001	010001 1110
D30.6	110 11110	011110 0110	100001 0110	D30.7	111 11110	011110 0001	100001 1110
D31.6	110 11111	101011 0110	010100 0110	D31.7	111 11111	101011 0001	010100 1110

Table 23 - Valid Special Characters		
Special Code Name	Current RD - abcdei fghj	Current RD + abcdei fghj
K28.0	001111 0100	110000 1011
K28.1	001111 1001	110000 0110
K28.2	001111 0101	110000 1010
K28.3	001111 0011	110000 1100
K28.4	001111 0010	110000 1101
K28.5	001111 1010	110000 0101
K28.6	001111 0110	110000 1001
K28.7	001111 1000	110000 0111
K23.7	111010 1000	000101 0111
K27.7	110110 1000	001001 0111
K29.7	101110 1000	010001 0111
K30.7	011110 1000	100001 0111

Unless a transmitter is operating in diagnostic mode, th K28.7 Special Character shall not be followed by any of the following Special or Data Characters: K28.x, D3.x, D11.x, D12.x, D19.x, D20.x, or D28.x, where x is a value in the range 0 to 7, inclusive (see clause 14 for a discussion of diagnostic mode).

NOTE - The above restriction simplifies receiver word synchronization hardware.

11.3 Word encoding and decoding

A Transmission Word is composed of four contiguous Transmission Characters treated as a unit. A word prior to transmission and after reception is composed of a combination of four Valid Data Bytes and Special Codes treated as a unit. These Valid Data Bytes and Special Codes shall be encoded according to the rules specified by 11.2 to create a Transmission Word. Likewise, the Transmission Characters of a Transmission Word shall be decoded according to the rules specified by 11.2 to create Valid Data Bytes and Special Codes.

11.4 Order d S ts

Tables 24, 25, and 26 specify the Ordered Sets (composed of Special and Data Characters) which are defined for use by this standard. The following Ordered Set types are defined in clause 16:

- Frame Delimiters
- Primitive Signals
- Primitive Sequences

Table 24 - Frame delimiters

Delimiter Function	Beginning RD	Ordered Set
SOF Connect Class 1 (SOFc1)	Negative	K28.5 D21.5 D23.0 D23.0
SOF Initiate Class 1 (SOFi1)	Negative	K28.5 D21.5 D23.2 D23.2
SOF Normal Class 1 (SOFn1)	Negative	K28.5 D21.5 D23.1 D23.1
SOF Initiate Class 2 (SOFi2)	Negative	K28.5 D21.5 D21.2 D21.2
SOF Normal Class 2 (SOFn2)	Negative	K28.5 D21.5 D21.1 D21.1
SOF Initiate Class 3 (SOFi3)	Negative	K28.5 D21.5 D22.2 D22.2
SOF Normal Class 3 (SOFn3)	Negative	K28.5 D21.5 D22.1 D22.1
SOF Fabric (SOFf)	Negative	K28.5 D21.5 D24.2 D24.2
EOF Terminate (EOFt)	Negative Positive	K28.5 D21.4 D21.3 D21.3 K28.5 D21.5 D21.3 D21.3
EOF Disconnect-Terminate (EOFdt)	Negative Positive	K28.5 D21.4 D21.4 D21.4 K28.5 D21.5 D21.4 D21.4
EOF Abort (EOFa)	Negative Positive	K28.5 D21.4 D21.7 D21.7 K28.5 D21.5 D21.7 D21.7
EOF Normal (EOFn)	Negative Positive	K28.5 D21.4 D21.6 D21.6 K28.5 D21.5 D21.6 D21.6
EOF Disconnect-Terminate-Invalid (EOFdti)	Negative Positive	K28.5 D10.4 D21.4 D21.4 K28.5 D10.5 D21.4 D21.4
EOF Normal-Invalid (EOFni)	Negative Positive	K28.5 D10.4 D21.6 D21.6 K28.5 D10.5 D21.6 D21.6

Explanation:

SOF - Start-of-frame delimiter
EOF - End-of-frame delimiter

Table 25 - Primitive Signals		
Primitive Signal	Beginning RD	Ordered Set
Idle	Negative	K28.5 D21.4 D21.5 D21.5
Receiver_Ready (R_RDY)	Negative	K28.5 D21.4 D10.2 D10.2

Table 26 - Primitive Sequences		
Primitive Sequence	Beginning RD	Ordered Set
Offline (OLS)	Negative	K28.5 D21.1 D10.4 D21.2
Not_Operational (NOS)	Negative	K28.5 D21.2 D31.5 D5.2
Link_Reset (LR)	Negative	K28.5 D9.2 D31.5 D9.2
Link_Reset_Response (LRR)	Negative	K28.5 D21.1 D31.5 D9.2

NOTES

1 Each EOF-delimiter Ordered Set is defined such that negative Current Running Disparity will result after processing of the final (rightmost) character of the Ordered Set. This, in combination with the Running Disparity Initialization rules specified in 11.2.2, ensures that the first Ordered Set following an EOF delimiter, transmitter power on, or transmitter exit from diagnostic mode will always be transmitted with negative Beginning Running Disparity. The Ordered Sets defined for the Primitive Signals and Primitive Sequences preserve this negative Disparity, ensuring that the Ordered Sets associated with SOF Delimiters, Primitive Signals, and Primitive Sequences will also always be transmitted with negative Beginning Running Disparity. As a result, Primitive Signal, Primitive Sequence, and SOF Delimiter Ordered Sets are defined for the negative Beginning Running Disparity case only. The primary benefit of such a definition is that it allows idle words to be removed and added from an encoded bit stream one word at a time without altering the Beginning Running Disparity associated with the Transmission Word subsequent to the removed idle word.

2 The K28.5 Special Character is chosen as the first character of all Ordered Sets for the following reasons:

— Bits abcdeif make up a Comma; this is a singular bit pattern which in the absence of transmission errors cannot appear in any other location of a Transmission Character and cannot be generated across the boundaries of any two adjacent Transmission Characters. The Comma can be used to easily find and verify character and word boundaries of the received bit stream.

— Bits ghj of the encoded character present the maximum number of transitions, simplifying receiver acquisition of bit synchronization.

3 The second character of the Ordered Sets used to represent EOF Delimiters differentiates between normal and invalid frames. It also ensures that the Running Disparity resulting after processing of an EOF Ordered Set is negative independent of the value of Beginning Running Disparity. Link_Reset (LR) and Link_Reset_Response (LRR) Ordered Sets are also differentiated through the use of their second characters. In all other Ordered Sets, it serves only as a placeholder which provides a high number of bit transitions.

4 The third and fourth characters of the Delimiter functions, Receiver_Ready, and the idle word are repeated to ensure that an error affecting a single character will not result in the recognition of an Ordered Set other than the one transmitted. The third and fourth characters of the other Ordered Sets defined by the standard have been selected to provide distinct patterns which provide a large number of transitions and good spectral characteristics.

5 The categories of Delimiter, Primitive Signal, and Primitive Sequence have meaning to the port and are defined by clause 16. Transmitters and receivers recognize the character combinations defined in this section as Ordered Sets only.

12 FC-1 receiver and transmitter description

12.1 Receiver state description

Whenever a signal (as defined in 5.6) is detected on a fibre and the receiver is not in Loopback mode, the receiver attached to that fibre shall attempt to achieve synchronization on both bit and Transmission-Word boundaries of the received encoded bit stream (see clause 13 for a description of Loopback mode). A definition of Bit Synchronization is provided in clause 3. Transmission-Word synchronization is defined in 12.1.2.2. Synchronization failures on either bit or Transmission-Word boundaries are not separately identifiable and cause loss-of-synchronization errors.

12.1.1 Receiver states

The receiver state diagram is shown in figure 42.

12.1.1.1 Operational states

Synchronization to Transmission-Word boundaries, hereafter referred to as *Synchronization*, is achieved when the receiver identifies the same Transmission-Word boundary on the received bit stream as that established by the transmitter at the other end of the fibre. The procedure used to achieve this condition is described in 12.1.2.2. When this condition is achieved, the receiver shall enter the *Synchronization-Acquired* state. A receiver in the Synchronization-Acquired state shall provide information that has been received from its attached fibre and decoded.

When the Transmission-Word boundary identified on the received bit stream by the receiver no longer matches the boundary established by the transmitter to which it is connected, the receiver shall enter the *Loss-Of-Synchronization* state. Such a receiver shall remain Operational but shall cease to provide received and decoded information. When the receiver is in the Loss-Of-Synchronization state, the procedure described in 12.1.2 shall be used to regain Synchronization.

When the receiver is Operational, it is always in one of th two states described above. The determination of an Operational receiver's state is based on th conditions described in 12.1.2 and 12.1.3.

12.1.1.2 Not Operational state

A receiver shall become Not Operational and shall enter the *Reset* state when a reset condition is imposed upon it, either internally or externally.

12.1.2 Entry into Synchronization-Acquired state

A receiver shall enter the Synchronization-Acquired state when it has achieved both bit and Transmission-Word synchronization.

12.1.2.1 Bit Synchronization

An Operational receiver that is in the Loss-Of-Synchronization state shall first acquire Bit Synchronization before attempting to acquire Transmission-Word synchronization. A definition of Bit Synchronization is provided in clause 3. After achieving Bit Synchronization, the receiver shall remain in the Loss-Of-Synchronization state until it achieves Transmission-Word synchronization.

12.1.2.2 Transmission-Word synchronization

An Operational receiver that is in the Loss-Of-Synchronization state and that has acquired Bit Synchronization shall attempt to acquire Transmission-Word synchronization. Transmission-Word synchronization is acquired by the detection of three Ordered Sets containing Commas in their leftmost bit positions without an intervening invalid Transmission Word, as specified by "Invalid Transmission Word rules." Upon acquisition of Transmission-Word synchronization, the receiver shall enter the Synchronization-Acquired state. The third detected Ordered Set shall be considered valid information and shall be decoded and provided by the receiver to its port. Ordered Set detection shall include both detection of the individual characters that make up an Ordered Set and proper alignment of those characters (i.e., the Special Character used to designate an Ordered Set shall be aligned in the leading (leftmost) character position of the received Transmission Word).

Word alignment of the received bit stream shall be achieved via the detection of a Comma or K28.5 Transmission Character in that stream. In the absence of bit errors, placement of a detected Comma or K28.5 Transmission Character at the leftmost position of a received Transmission Word ensures proper alignment of that word and of subsequently received Transmission Words.

While attempting to achieve word synchronization, an Operational receiver shall be required to detect only those Ordered Sets which contain a Comma in their leftmost bit positions.

The method used by the receiver to implement the word alignment function and to detect Ordered Sets is not defined by this International Standard.

NOTES

1 The Comma contained within the K28.1, K28.5, and K28.7 Special Characters is a singular bit pattern which in the absence of transmission errors cannot appear in any other location of a Transmission Character and cannot be generated across the boundaries of any two adjacent Transmission Characters. This bit pattern is sufficient to identify the word alignment of the received bit stream. Some implementations (e.g., those which choose to implement the word alignment function in continuously-enabled mode) may choose to align on the full K28.5 Ordered Set to decrease the likelihood of false alignment when bit errors are present in the received bit stream.

2 The electrical interface described in annex D allows the word alignment function to be enabled in either of two modes:

- Continuously-enabled word alignment (continuous-mode alignment)
- Explicitly-enabled word alignment (explicit-mode alignment)

Continuous-mode alignment allows the receiver to reestablish word alignment at any point in the incoming bit stream while the receiver is Operational. Such realignment is likely (but not guaranteed) to result in Code Violations and subsequent loss of Synchronization. Under certain conditions, it may be possible to realign an incoming bit stream without loss of Synchronization. If such a realignment occurs within a received frame, detection of the resulting error condition is dependent upon higher-level function (e.g., invalid CRC, missing EOF Delimiter).

Explicit-mode alignment allows the receiver to reestablish word alignment under controlled circumstances (e.g., while in the Loss-Of-Synchronization state). Once Synchronization has been acquired, the word alignment function of the receiver is disabled.

12.1.3 Entry into Loss-Of-Synchronization state

The following four conditions shall cause an Operational receiver to enter the Loss-Of-Synchronization state:

- Completion of the Loss-Of-Synchronization procedure
- Transition to power on
- Exit from receiver reset condition
- Detection of loss of signal

NOTE - While in the Loss-Of-Synchronization state, the receiver may attempt to reacquire bit resynchronization. In some instances, this may allow the receiver to regain Synchronization and enter the Synchronization-Acquired state when it otherwise would not be possible. However, initiation of bit resynchronization may also delay the resynchronization process by forcing the receiver to reestablish a clock reference when such reestablishment is otherwise unnecessary. See 5.5 and D.5.1 for a detailed discussion of Bit Synchronization.

12.1.3.1 Loss-of-Synchronization procedure

The loss-of-Synchronization procedure defines the method by which the receiver changes from the Synchronization-Acquired state to the Loss-Of-Synchronization state. The procedure tests each received Transmission Word according to the rules defined in "Invalid Transmission Word rules" to determine the validity of the Transmission Word.

Starting in the Synchronization-Acquired state, the receiver shall check each received Transmission Word according to the rules of "Invalid Transmission Word rules" to determine if the word is valid. The receiver shall continue to check the Transmission Words and shall remain in the Synchronization-Acquired state until the loss-of-Synchronization procedure, as described by "Loss-of-Synchronization-procedure states," is completed.

Invalid Transmission Word rules

An invalid Transmission Word shall be recognized by the receiver when one of th following conditions is detect d:

- A Code Violation on a character basis, as specified by tables 22 and 23, is detected

within a Transmission Word. This is referred to as a Code Violation condition.

— Any valid Special Character is detected in the second, third, or fourth character position of a Transmission Word. This is referred to as an invalid Special Code alignment condition.

— A defined Ordered Set (as listed in 11.4) is received with improper Beginning Running Disparity (e.g., an SOF delimiter is received with positive Beginning Running Disparity, an EOF delimiter specified for positive Beginning Running Disparity is received when Beginning Running Disparity for that Transmission Word is negative). This is referred to as an invalid Beginning Running Disparity condition.

Loss-of-Synchronization-procedure states.
The following five detection states are defined as part of the loss-of-Synchronization procedure:

a) No invalid Transmission Word has been detected (the No-Invalid-Transmission-Word detection state).

b) The first invalid Transmission Word is detected (the First-Invalid-Transmission-Word detection state).

c) The second invalid Transmission Word is detected (the Second-Invalid-Transmission-Word detection state).

d) The third invalid Transmission Word is detected (the Third-Invalid-Transmission-Word detection state).

e) The fourth invalid Transmission Word is detected (the Fourth-Invalid-Transmission-Word detection state).

A receiver in the Synchronization-Acquired state may be in any of the first four detection states listed above. A receiver in the fifth detection state listed above (the Fourth-Invalid-Transmission-Word detection state) shall enter the Loss-Of-Synchronization state.

When the procedure is in detection state a, checking for an invalid Transmission Word shall be performed. After each invalid Transmission Word is detected, one of the detection states b through e shall be entered. When the procedure is in detection state b, c, or d, checking for additional invalid Transmission Words shall be performed in groups of two consecutive Transmission Words. If two consecutive valid Transmission Words are received, the count of previously detected invalid Transmission Words shall be reduced by one, and the previous detection state shall be entered. The loss-of-Synchronization procedure is completed when detection state e is entered.

The No-Invalid-Transmission-Word detection state shall be entered on a transition to the Synchronization-Acquired state or after the First-Invalid-Transmission-Word detection state is reset.

The First-Invalid-Transmission-Word detection state shall be entered after the first invalid Transmission Word is detected or after the previous detection state (detection of the second invalid Transmission Word) is reset. Subsequent Transmission Words received shall be checked to determine whether an additional invalid Transmission Word is detected within the next consecutive two or fewer Transmission Words received. If an additional invalid Transmission Word is detected within the next consecutive two or fewer Transmission Words received, the Second-Invalid-Transmission-Word detection state shall be entered. If two consecutive valid Transmission Words are received, the No-Invalid-Transmission-Word detection state shall be entered.

The Second-Invalid-Transmission-Word detection state shall be entered after the second invalid Transmission Word is detected or after the previous detection state (detection of the third invalid Transmission Word) is reset. Subsequent Transmission Words received shall be checked to determine whether an additional invalid Transmission Word is detected within the next consecutive two or fewer Transmission Words received. If an additional invalid Transmission Word is detected within the next consecutive two or fewer Transmission Words received, the Third-Invalid-Transmission-Word detection state shall be entered. If two consecutive valid Transmission Words are received, the First-Invalid-Transmission-Word detection state shall be entered.

The Third-Invalid-Transmission-Word detection state shall be entered after the third invalid Transmission Word is detected. Subsequent Transmission Words received shall be checked to determine whether an additional invalid Transmission Word is detected within the next consecutive two or fewer Transmission Words received. If an additional invalid Transmission

Word is detected within the next consecutive two or fewer Transmission Words received, the Fourth-Invalid-Transmission-Word detection state shall be entered and the receiver shall immediately enter the Loss-Of-Synchronization state. If two consecutive valid Transmission Words are received, the Second-Invalid-Transmission-Word detection state shall be entered.

The Fourth-Invalid-Transmission-Word detection state shall be entered after the fourth invalid Transmission Word is detected. Upon entering this detection state, the receiver shall immediately enter the Loss-Of-Synchronization state. Subsequent Transmission Words received shall not be checked as the loss-of-Synchronization procedure is complete. The receiver shall remain in the Loss-Of-Synchronization state until one of the following occurs:

— The receiver regains Synchronization

— The receiver is reset

The following figure graphically portrays the Loss-of-Synchronization procedure. States a through e are keyed to the detection states described by the ordered list at the beginning of this section. State a is the initial detection state entered by a receiver upon acquisition of Synchronization. States a through d are detection states which are possible only when the receiver has achieved Synchronization. Entry into State e results in loss of receiver Synchronization.

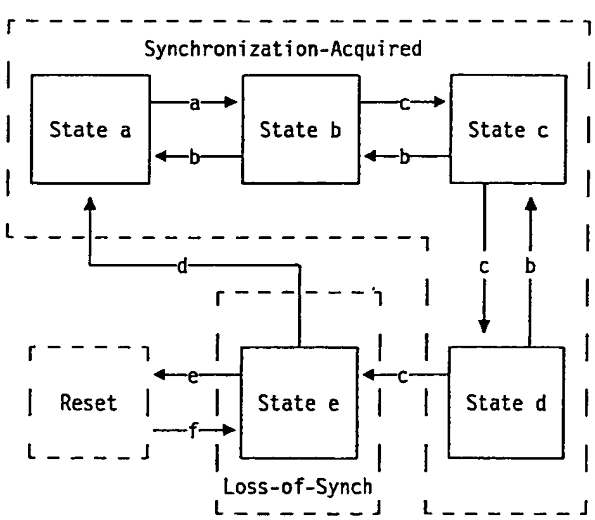


Figure 41 - L ss-of-Synchronization pr cedur state diagram

State Transition Conditions:

a) The first invalid Transmission Word is detected

b) An additional invalid Transmission Word is not detected in the next two or fewer consecutive Transmission Words

c) An additional invalid Transmission Word is detected in the next two or fewer consecutive Transmission Words

d) The receiver regains Synchronization

e) The receiver is Reset

f) The receiver exits a previously established Reset condition

NOTE - The rationale for the loss-of-Synchronization procedure is to reduce the likelihood that a single error will result in a loss of Synchronization. For example, a single two-bit error positioned to overlap two Transmission Words could result in the detection of three invalid Transmission Words; the two Transmission Words directly affected by the error and a subsequent Transmission Word which was affected by a disparity change resulting from the error. The procedure described above would maintain Synchronization in such a case.

12.1.3.2 Transition to power on

When the receiver is Operational after being powered on, the receiver shall enter the Loss-Of-Synchronization state.

12.1.3.3 Exit from receiver reset condition

When the receiver is Operational after exiting from a receiver reset condition imposed upon it, either externally or internally, the receiver shall enter the Loss-Of-Synchronization state.

NOTE - The conditions required for a receiver in the Reset state to exit that state are not defined by this International Standard. Such conditions may be based on explicit indications. They may also be time-dependent in nature.

12.1.3.4 Detection of loss of signal

When a loss-of-signal condition is recognized by an Operational receiver, the Loss-Of-Synchronization state shall be entered (if the receiver is not presently in that state). The receiver shall remain in this state until one of the following conditions occur:

— The loss-of-signal condition is corrected and Synchronization is regained

— The receiver is reset

12.1.4 Entry into Reset state

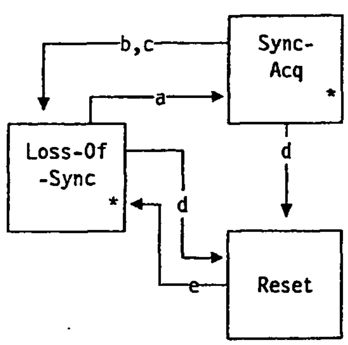
When a receiver reset condition is imposed on a receiver, either internally or externally, the receiver shall enter the Reset state. Once the Reset state is entered, the receiver shall become Not Operational and shall remain in the Reset state until it is subsequently made Operational by exiting the receiver reset condition.

NOTES

1 A typical use of receiver reset is to force a receiver in the Loss-Of-Synchronization state to attempt reacquisition of Bit Synchronization (note that entry into this state does not necessarily indicate loss of Bit Synchronization).

2 The conditions required for a receiver in the Reset state to exit that state are not defined by this International Standard. Such conditions may be based on explicit indications. They may also be time-dependent in nature.

12.2 Receiver state diagram



* See note 3 on page 76.

Figure 42 - Receiver state diagram

State Transition Conditions:

a) Acquisition of Synchronization (see 12.1.2)

b) Completion of loss-of-Synchronization procedure (see 12.1.3.1)

c) Detection of a loss of signal condition (see 12.1.3.4)

d) Reset condition imposed on the receiver (see 12.1.4)

e) Exiting of receiver reset condition (see 12.1.3.3)

NOTES

1 The Loss-Of-Synchronization state is the default state upon completion of receiver Initialization.

2 The data path width of a variable-path-width receiver must be established before that receiver is capable of entering the Synchronization-Acquired state.

3 The receiver may be in either Loopback or normal mode when in states denoted by * (see clause 13 for a discussion of Loopback mode).

12.3 Transmitter state description

While Operational and not disabled, a transmitter attached to a Fibre shall constantly attempt to transmit an encoded bit stream onto that Fibre. Some transmitters are capable of monitoring this transmitted signal and verifying its validity. Such transmitters may become Not Operational as a result of this monitoring. A transmitter may also be placed in a disabled state under certain internal or external conditions.

The transmitter state diagram is shown in figure 43.

12.3.1 Transmitter states

12.3.1.1 Operational states

A transmitter actively attempting to transmit an encoded bit stream onto its attached fibre is defined to be in the *Working* state.

Under certain conditions, it may be necessary or desirable to cease the transmission of signals by a transmitter onto its attached Fibre. When such an action is initiated by a request from the port associated with a transmitter or as the result of an external event, a transmitter shall enter the *Not-Enabled* state. When such an action results from detection of a laser safety condition, a transmitter shall enter the *Open-Fibre* state. A transmitter in a Not-Enabled or Open-Fibre state shall remain Operational.

NOTE - A transmitter in a Not-Enabled or Open-Fibre state is free to transmit those signals associated with laser safety procedures. See clause 15 and 6.2.3 for additional information related to laser safety.

12.3.1.2 Not Operati nal state

A transmitter shall become Not Operational and shall enter the *Failure* state when a failure condition is detected by the transmitter. Detectable transmitter failure conditions are defined by vendor unique signal monitoring rules. Compliance with the standard shall not be affected by the presence or absence of such rules in an implementation.

A transmitter-failure condition shall be recognized and reported by the transmitter to its associated port when the transmitter becomes Not Operational and enters the Failure state.

12.3.2 Entry into Working state

An Operational transmitter shall enter the Working state when its transmitter becomes enabled. A transmitter shall become enabled under either of the following conditions:

- Processing of a port request to enter the enabled state when the transmitter was not previously placed in the Open-Fibre state as a result of laser safety procedures
- Removal of a previously-detected laser safety condition (i.e., a condition which requires that the transmitter cease transmission) while no transmitter disable request remains outstanding

While in the Working state, a transmitter shall actively attempt to transmit valid requests for information transfer from its associated port. Valid requests for information transfer conform to the following rules:

- Requests for Ordered Set transmission are aligned with the word alignment established by the transmitter upon entry into the Working state.
- Requests for Valid Data Byte transmission are aligned with the byte alignment established by the transmitter upon entry into the Working state. Each byte of a four-byte word is aligned based on the established word alignment.

The following requests for information transfer are inconsistent with the established transmission rules and cannot occur:

- Transmission requests improperly aligned with transmission (word and byte) boundaries
- Transmission requests not consistent with the current transmission boundary
  - Requests for Ordered Set transfers on a non-word boundary
  - Requests for data transfers overlapping an Ordered Set transfer currently in progress
- Lack of transmission requests (i.e., additional bytes required to complete a Transmission Word) when a Transmission Word has been partially transmitted as a result of prior transmission requests

12.3.3 Entry into Not-Enabled state

Entry into the Not-Enabled state may result from external error conditions detected by the Link Control Facility, or it may result from internally generated signals. Specific conditions under which the transmitter shall enter the Not-Enabled state are as follows:

- Completion of transmitter Initialization
- Processing of a port request to enter the Not-Enabled state when a laser safety condition is not currently present
- Removal of a previously-detected laser safety condition while a transmitter disable request remains outstanding

The transmitter shall remain in the Not-Enabled state until the conditions causing it to cease transmission are removed.

12.3.4 Entry into Open-Fibre state

Entry into the Open-Fibre state results from external error conditions detected by the Link Control Facility. Specifically, the transmitter enters the Open-Fibre state upon determination by FC-0 OFC that a laser safety condition exists.

The transmitter shall remain in the Open-Fibre state until the laser safety condition is removed. A state change shall not result from receipt of a transmitter disable or nable request while in the Open-Fibre stat . However, such receipt may affect the subsequent state change which results upon removal of the previously-detected laser safety condition (see 12.3.2 and 12.3.3).

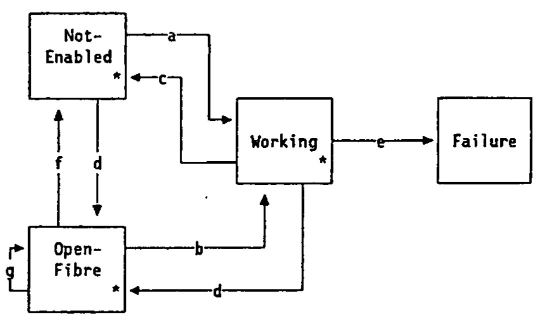
12.3.5 Entry int Failur state

A transmitter shall enter into the Failure state upon recognition (via signal monitoring) of a transmitter-failure condition while in the Working state. Once the Failure state is entered, the transmitter shall become Not Operational and shall remain in the Failure state until it is subsequently made Operational through external intervention.

A transmitter shall not be required to provide a signal monitoring function. Each transmitter capable of monitoring its transmitted signal shall define a method by which this monitoring is to take place. It shall also define the conditions necessary to cause the transmitter to recognize a transmitter-failure condition and change from the Working state to the Failure state.

NOTE - Monitoring of average light levels is a typical method of transmitter monitoring. Typically, an error condition is detected when the transmitted light level of a transmitter falls outside of the range of allowed values specified for its transmitter class (see 5.2 for a discussion of signal monitoring.).

12.4 Transmitter state diagram



* See note 3 on page 78.

Figure 43 - Transmitter state diagram

State Transition Conditions:

a) Receipt of transmitter enable request from the port when a laser safety condition is not currently present (see 12.3.2)

b) Determination that a laser safety condition no longer exists and no transmitter disable request remains outstanding (see 12.3.2)

c) Receipt of transmitter disable request from the port when a laser safety condition is not currently present (see 12.3.3)

d) Determination that a laser safety condition exists (see 12.3.4)

e) Detection of a transmitter failure condition (see 12.3.5)

f) Determination that a laser safety condition no longer exists when an outstanding transmitter disable request exists (see 12.3.3)

g) Receipt of transmitter disable or enable request (see 12.3.4)

NOTES

1 The Not_Enabled state is the default state upon completion of transmitter Initialization.

2 The data path width of a variable-path-width transmitter must be established before that transmitter may enter the Working state.

3 The transmitter may be in either Loopback or normal mode when in states denoted by * (see clause 13 for a discussion of Loopback mode).

13 Loopback mode

Loopback mode shall be provided by the transmitter and receiver of a port as a diagnostic function to the port. When Loopback mode is selected, transmission requests passed to the transmitter shall be shunted directly to the receiver, overriding any signal detected by the receiver on its attached Fibre. A Link Control Facility shall be explicitly placed in Loopback mode (i.e., Loopback mode is not the normal mode of operation of a Link Control Facility). The method of implementing Loopback mode is not defined by this standard.

NOTE - Loopback mode may be implemented either in the parallel or the serial circuitry of a Link Control Facility.

13.1 Receiver considerations

A receiver in the Loss-Of-Synchronization or Synchronization-Acquired state may be placed in Loopback mode. A receiver in any other state shall not be placed in Loopback mode.

Entry into or exit from Loopback mode may result in a temporary loss of Synchronization. Under such conditions, decoded information shall not be presented by the receiver to the port until Synchronization has been reestablished.

13.2 Transmitter considerations

A transmitter in the Working, Open-Fibre, or Not-Enabled state may be placed in Loopback mode. A transmitter in any other state shall not be placed in Loopback mode. The external behavior of a transmitter (i.e., the activity of a transmitter with respect to its attached Fibre) simultaneously in the Working state and in Loopback mode is unpredictable.

14 Diagnostic mode

A limited set of diagnostic functions may be provided as an implementation option for testing of the transmitter function of the FC-0 portion of the Fibre Channel (see annex B).

Diagnostic functions
Some diagnostic functions which are not defined by this standard may be provided by certain implementations. Compliance with the standard shall not be affected by the provision or exclusion of such functions by an implementation.

NOTE - A typical diagnostic function is the ability to transmit invalid Transmission Characters within an otherwise valid bit stream. Certain invalid bit streams may cause a receiver to lose word and/or bit synchronization. See 5.4 for a more detailed discussion of receiver and transmitter behavior under various diagnostic conditions.

15 Transmitter safety

Short wave (780nm) laser transmitters (as defined in 6.2) require special procedures to ensure that Class 1 laser safety standards are met. These procedures have the following effect on the transmitter definition:

- A special transmitter state (the Open-Fibre state) is defined.
- When FC-0 OFC determines that a laser safety condition (i.e., a condition which requires that the transmitter cease transmission) exists, the transmitter shall enter the Open-Fibre state.
- When FC-0 OFC determines that a laser safety condition no longer exists, the transmitter shall exit the Open-Fibre state.
- Port requests are incapable of forcing a transmitter which enters the Open-Fibre state as a result of FC-0 OFC to exit that state.

For a detailed description of FC-0 OFC, see 6.2.3 and 6.1.

16 Ordered Sets

16.1 Introduction

An Ordered Set is a four-character combination of data and special transmission characters. FC-PH designates a number of Ordered Sets to have special meaning. Ordered Sets provide the ability to obtain bit and word synchronization which also establishes word boundary alignment. See 11.4 for additional information on Ordered Sets and 12.1.2.2 on rules for synchronization. The following types of Ordered Sets are defined:

- Start_of_Frame delimiters
- End_of_Frame delimiters
- Primitive Signals
  - Idle
  - Receiver_Ready (R_RDY)
- Primitive Sequences
  - Not_Operational (NOS)
  - Offline (OLS)
  - Link_Reset (LR)
  - Link_Reset_Response (LRR).

If an unrecognized Ordered Set is detected, it shall be treated as an Idle Primitive Signal.

NOTE - Treating unrecognized Ordered Sets as Idles allows future introduction of Ordered Sets for additional features and functions beyond the scope of FC-PH.

16.2 Frame delimiters

A frame delimiter is an Ordered Set that immediately precedes or follows the contents of a frame. Separate and distinct delimiters shall identify the start of a frame and the end of a frame. Frame delimiters shall be recognized when a single Ordered Set is detected. See 11.4 for details on Ordered Sets and table 24 for the bit encodings for delimiters.

16.2.1 Start_of_Frame (SOF)

The Start_of_Frame (**SOF**) delimiter is an Ordered Set that immediately precedes the frame content. There are multiple **SOF** delimiters defined for th Fabric and for N_Port Sequence control. The **SOF** delimiter shall be transmitted on a word boundary. See 17.1 for more information on fram transmission.

16.2.2 End_of_Frame (EOF)

The End_of_Frame (**EOF**) delimiter is an Ordered Set that shall immediately follow the CRC and shall be transmitted on a word boundary. The **EOF** delimiter shall designate the end of the frame content and shall be followed by Idles. There are multiple **EOF** delimiters defined for the Fabric and for N_Port Sequence control. See 17.1 for more information on frame transmission.

16.3 Primitive Signals

A Primitive Signal is an Ordered Set designated by FC-PH to have special meaning. Primitive Signals shall be recognized when a single Ordered Set is detected. See 11.4 for details on Ordered Sets and table 25 for the bit encodings for Primitive Signals.

16.3.1 Idle

An Idle is a Primitive Signal transmitted on the link to indicate an operational Port facility ready for frame transmission and reception. Idles shall be transmitted on the link during those periods of time when frames, R_RDY, or Primitive Sequences are not being transmitted. See 17.1 regarding frame transmission and 17.8 regarding frame reception.

16.3.2 Receiver_Ready (R_RDY)

The R_RDY Primitive Signal shall indicate that a single Class 1 connect-request (**SOFc1**), Class 2, or Class 3 frame was received and that the interface buffer which received the frame is available for further frame reception. Validity of the frame content shall not be required. Transmission of the R_RDY Primitive Signal shall be preceded and followed by a minimum of two Idles and at the N_Port transmitter, there shall be a minimum of six Primitive Signals (R_RDY and Idles) between frames. The R_RDY Primitive Signal shall be received by the F_Port but shall not b passed through the Fabric, if present. Since R_RDY is not passed through the Fabric, there is

no requirement for additional Idles for clock skew management (see 17.1). See 20.3.2.1 for a discussion regarding th use of R_RDY in flow control in conjunction with ACK frames.

16.4 Primitive Sequences

A Primitive Sequence is an Ordered Set that is transmitted repeatedly and continuously. Primitive Sequences are transmitted to indicate specific conditions within a Port (N_Port or F_Port) or conditions encountered by the Receiver logic of a Port. See table 26 for bit encodings of Primitive Sequences.

Primitive Sequences shall be transmitted continuously while the condition exists. A detailed description of Port state changes relative to Primitive Sequence reception and transmission is shown in Q.4. When a Primitive Sequence is received and recognized, depending on the State of the Port, a corresponding Primitive Sequence or Idles shall be transmitted in response. The following Primitive Sequence Protocols are described:

- Link Initialization (see 16.6.2)
- Online to Offline (see 16.6.3)
- Link Failure (see 16.6.4)
- Link Reset (see 16.6.5).

Primitive Sequences transmitted from an N_Port to its locally attached F_Port of a Fabric shall remove a pending or existing Dedicated Connection. The locally attached F_Port shall respond appropriately to the Primitive Sequence received and shall notify the F_Port attached to the other connected N_Port which shall transmit the Link Reset Primitive to the other connected N_Port (i.e., initiate Link Reset Protocol).

If a Dedicated Connection does not exist, a Primitive Sequence transmitted by an N_Port shall be received and recognized by the locally attached F_Port, but not transmitted through the Fabric.

16.4.1 Primitive Sequence Recognition

Recognition of a Primitive Sequence shall require consecutive detection of three instances of the same Ord red Set without any intervening data indications from the Receiver logic (FC-1).

16.4.2 N t_Operational (NOS)

The Not_Operational Primitive Sequence shall be transmitted to indicate that the Port transmitting this Sequence has detected a Link Failure condition or is Offline, waiting for OLS to be received.

Link failure conditions shall be defined as:

- Loss of Synchronization for more than a timeout period (R_T_TOV) while not in the Offline State (see 12.1.1.2),
- Loss of Signal while not in the Offline State (see 12.1.1.2),
- Timeout (R_T_TOV) during the Link Reset Protocol (see 16.6.5).

Note - Loss of Signal is an optional indication related to FC-0 implementations which have the ability to detect the presence of an input (see 6.2.3.2 for Loss of Light and H.10 for FC-0 signal_detect.indication).

16.4.3 Offline (OLS)

The Offline Primitive Sequence shall be transmitted to indicate that the Port transmitting this Sequence is:

- initiating the Link Initialization Protocol,
- receiving and recognizing NOS, or
- entering the Offline State.

A Port shall transmit the OLS Primitive Sequence for a minimum period of 5 ms before further actions are taken. A Port shall enter the Offline State in order to perform diagnostics or power-down.

NOTE - The value of 5 ms is based on distances up to 10 km. If longer distances are employed in the future, the time should be increased accordingly.

16.4.4 Link Reset (LR)

The Link Reset Primitive Sequence shall be transmitted by a Port to initiate the Link Reset Protocol (see 16.6.5) or to recover from a Link Timeout (see 29.2.3). An N_Port supporting Class 1 Service may also transmit the LR Primitive Sequence when it is unable to determine its Connection status, a procedure known as Connection Recov ry (see 28.8).

16.4.5 Link Reset Response (LRR)

The Link Reset Response Primitive Sequence shall be transmitted by a Port to indicate that it is receiving and recognizes the LR Primitive Sequence.

16.5 Port states

This section defines the possible states for a Port. For conditions which are not explicitly listed to cause state changes to occur, the Port shall remain within the current state.

16.5.1 Active State (AC)

A Port shall enter the Active State when it completes the Link Initialization Protocol (see 16.6.2) or the Link Reset Protocol (see 16.6.5). When a Port is in the Active State, it is able to transmit and receive frames and Primitive Signals (Idle and R_RDY). When a Primitive Sequence is received, the Port exits the Active State and shall enter another state based on the Primitive Sequence received:

– If LR is received and recognized, enter the LR Receive State (see 16.5.2.2).

– If LRR is received and recognized, enter the LRR Receive State (see 16.5.2.3). A Primitive Sequence protocol error is detected and counted in the LESB (see 29.8).

– If NOS is received and recognized, enter the NOS Receive State (see 16.5.3.2).

– If OLS is received and recognized, enter the OLS Receive State (see 16.5.4.2).

– If Idles are received and recognized, remain in the Active State.

– If R_RDY is received and recognized, remain in the Active State.

– If a frame is received, remain in the Active State.

If certain conditions are detected within the Port, the Port may exit the Active State and perform a Primitive Sequence Protocol:

– Link Initialization (see 16.6.2)
– Online to Offline (see 16.6.3)
– Link Failure (see 16.6.4)
– Link Reset (see 16.6.5).

16.5.2 Link R covery State

When a Port is in Link Recovery State, it is in one of the following three substates.

16.5.2.1 LR Transmit State (LR1)

A Port shall enter the LR Transmit State to initiate the Link Reset Protocol. An N_Port supporting Class 1 may also enter the LR Transmit State when it is unable to determine its Connection status. While in the LR Transmit State, the Port shall transmit the LR Primitive Sequence.

a) While in the LR Transmit State, the Port shall respond as follows:

– If LR is received and recognized, enter the LR Receive State (see 16.5.2.2).

– If LRR is received and recognized, enter the LRR Receive State (see 16.5.2.3).

– If NOS is received and recognized, enter the NOS Receive State (see 16.5.3.2).

– If OLS is received and recognized, enter the OLS Receive State (see 16.5.4.2).

– If Idle or R_RDY is received and recognized, remain in the LR Transmit State.

b) If a Port remains in the LR Transmit State for a period of time greater than a timeout period (R_T_TOV), a Link Reset Protocol Timeout shall be detected which results in a Link Failure condition (enter the NOS Transmit State, see 16.5.3.1).

c) If a Loss of Signal is detected, the Port shall enter the NOS Transmit State (see 16.5.3.1). Loss of synchronization need not be checked since item b will occur first.

Transmission of the Link Reset Primitive Sequence has different consequences based on Class:

Class 1

In Class 1, a pending or existing Dedicated Connection shall be removed and the end-to-end Credit associated with the connected N_Ports shall be reset to the Login value. The locally attached F_Port shall enter the LR Receive State and shall notify the F_Port attached to the other connected N_Port which shall transmit the Link

Reset Primitive Sequence to the other connected N_Port (i.e., initiate Link Reset Protocol).

All Class 1 frame Sequences which are Active or Open in an N_Port when LR is received and recognized, or transmitted shall be abnormally terminated. When the Link Reset Primitive is being transmitted by an N_Port, all Class 1 frames received shall be discarded. See 28.8 for more discussion on Connection Recovery.

Class 2 and 3

Buffer-to-buffer Credit (associated with Class 1 (**SOFc1**), Class 2, and Class 3) within the N_Port or F_Port shall be reset to its Login value and an F_Port shall process or discard any Class 1 connect-request, Class 2, or Class 3 frames currently held in the receive buffer associated with the outbound fibre of the attached N_Port. Class 2 end-to-end Credit shall not be affected.

16.5.2.2 LR Receive State (LR2)

A Port shall enter the LR Receive State when it receives and recognizes the LR Primitive Sequence while it is not in the Wait for OLS or NOS Transmit State. While in the LR Receive State, the Port shall transmit the LRR Primitive Sequence.

a) While in the LR Receive State, the Port shall respond as follows:

   – If LR is received and recognized, remain in the LR Receive State.

   – If LRR is received and recognized, enter the LRR Receive State (see 16.5.2.3).

   – If NOS is received and recognized, enter the NOS Receive State (see 16.5.3.2).

   – If OLS is received and recognized, enter the OLS Receive State (see 16.5.4.2).

   – If Idles are received and recognized, enter the Active State (see 16.5.1).

b) If a Port remains in the LR Receive State for a period of time greater than a timeout period (R_T_TOV), a Link Reset Protocol Timeout shall be detected which results in a Link Failure condition (enter the NOS Transmit State, see 16.5.3.1).

c) If a Loss of Signal is det cted, the Port shall enter the NOS Transmit State (see 16.5.3.1). Loss of Synchronization need not be checked since item b will occur first.

Reception of the Link Reset Primitive Sequence has different consequences based on Class:

Class 1

In Class 1, a pending or existing Dedicated Connection shall be removed and the end-to-end Credit associated with the connected N_Ports shall be reset to the Login value. All Class 1 frame Sequences which are Active or Open in an N_Port when LR is received and recognized shall be abnormally terminated.

Class 2 and 3

A Port which receives and recognizes the Link Reset Primitive Sequence shall process or discard Class 2, or Class 3 frames currently held in its receive buffers. Buffer-to-buffer Credit (associated with Class 1 (**SOFc1**), Class 2, and Class 3) within the N_Port or F_Port shall be reset to its Login value.

16.5.2.3 LRR Receive State (LR3)

A Port shall enter the LRR Receive State when it receives and recognizes the LRR Primitive Sequence while it is in the Active, LR Transmit, LR Receive, or OLS Receive State. While in the LRR Receive State, the Port shall transmit Idles.

a) While in the LRR Receive State, the Port shall respond as follows:

   – If LR is received and recognized, enter the LR Receive State (16.5.2.2).

   – If LRR is received and recognized, remain in the LRR Receive State.

   – If NOS is received and recognized, enter the NOS Receive State (see 16.5.3.2).

   – If OLS is received and recognized, enter the OLS Receive State (see 16.5.4.2).

   – If Idles are received and recognized, the Port shall exit the LRR Receive State and enter the Active State (see 16.5.1).

b) If a Port r mains in the LRR Receive State for a period of time greater than a timeout period (R_T_TOV), a Link Reset Protocol Timeout shall be detected which results in a

Link Failure condition (enter the NOS Transmit State, see 16.5.3.1).

c) If a Loss of Signal is detected, the Port shall enter the NOS Transmit State (see 16.5.3.1). Loss of Synchronization need not be checked since item b will occur first.

NOTES

1 If a Port is in the NOS Transmit, NOS Receive, or OLS Transmit State, it ignores the LRR Primitive Sequence if it is received and recognized.

2 If a Port is in the Wait for OLS State, it enters the NOS Transmit State if the LRR Primitive Sequence is received and recognized.

16.5.3 Link Failure State

When a Port is in Link Failure State, it is one of the following substates.

16.5.3.1 NOS Transmit State (LF2)

A Port shall enter the NOS Transmit State when a Link Failure condition is detected. Upon entry into the NOS Transmit State, the Port shall update the appropriate error counter in the Link Error Status Block (see 29.8). Only one error per Link Failure event shall be recorded. The Port shall remain in the NOS Transmit State while the condition which caused the Link Failure exists. While in the NOS Transmit State, the Port shall transmit the NOS Primitive Sequence.

When the Link Failure condition is no longer detected, the Port shall respond as follows:

- If LR is received and recognized, remain in the NOS Transmit State.
- If LRR is received and recognized, remain in the NOS Transmit State.
- If NOS is received and recognized, enter the NOS Receive State (see 16.5.3.2).
- If OLS is received and recognized, enter the OLS Receive State (see 16.5.4.2).
- If Idle is received and recognized, remain in the NOS Transmit State.

Transmission of NOS by an N_Port to its locally attached F_Port of a Fabric shall remov a pending or existing Dedicated Connection. The locally attach d F_Port shall respond by entering the NOS Receive State and shall notify the F_Port attached to the other connected N_Port which shall transmit the Link Reset Primitive to the other connected N_Port (i.e., initiate Link Reset Protocol).

If a Dedicated Connection does not exist, NOS transmission by an N_Port shall be received and recognized by the locally attached F_Port, but not transmitted through the Fabric. The F_Port shall respond by entering the NOS Receive State.

16.5.3.2 NOS Receive State (LF1)

A Port shall enter the NOS Receive State when it receives and recognizes the NOS Primitive Sequence. Upon entry into the NOS Receive State, the Port shall update the appropriate error counter in the Link Error Status Block (see 29.8). Only one error per Link Failure event is recorded. While in the NOS Receive State, the Port shall transmit the OLS Primitive Sequence.

a) While in the NOS Receive State, the Port shall respond as follows:

   - If LR is received and recognized, enter the LR Receive State (see 16.5.2.2).
   - If LRR is received and recognized, remain in the NOS Receive State.
   - If NOS is received and recognized, remain in the NOS Receive State.
   - If OLS is received and recognized, enter the OLS Receive State (see 16.5.4.2).
   - If Idle is received and recognized, remain in the NOS Receive State.

b) If a Link Failure condition is detected due to Loss of Signal or Synchronization, the Port shall enter the NOS Transmit State (see 16.5.3.1).

c) A timeout (R_T_TOV) period is started when NOS is no longer recognized and no other events occur which cause a transition out of the NOS Receive State. If the timeout period expires, the Port shall enter the NOS Transmit State (see 16.5.3.1).

16.5.4 Offline Stat

While in the Offline State, a Port shall not r cord receiver errors such as loss of synchronization. A Port shall enter the Offline State under the following conditions:

– after power-up, or internal reset, before the Link Initialization Protocol is complete,

– after transmission of the first OLS Ordered Set, or

– after reception and recognition of the OLS Primitive Sequence.

When a Port is in the Offline state, it is in one of the following substates.

16.5.4.1 OLS Transmit State (OL1)

A Port shall enter the OLS Transmit State in order to:

– perform Link Initialization using the Link Initialization Protocol (see 16.6.2) in order to exit the Offline State.

– transition from Online to Offline using the Online to Offline Protocol (see 16.6.3).

When the Port enters the OLS Transmit State, it shall transmit OLS for a minimum time of 5 ms. After that period of time has elapsed, it shall proceed according to the steps defined by the Link Initialization Protocol or the Online to Offline Protocol based on the reason the Port entered the OLS Transmit State. While a Port is attempting to go Online, if no Primitive Sequence is received or event detected which causes the Port to exit the OLS Transmit State after a timeout period (R_T_TOV), the Port shall enter the Wait for OLS State.

Transmission of OLS by an N_Port to its locally attached F_Port of a Fabric shall remove a pending or existing Dedicated Connection. The locally attached F_Port shall respond by entering the OLS Receive State and shall notify the F_Port attached to the other connected N_Port which shall transmit the Link Reset Primitive Sequence to the other connected N_Port (i.e., initiate Link Reset Protocol).

If a Dedicated Connection does not exist, OLS transmission by an N_Port shall b received and recognized by the locally attached F_Port, but not transmitted through the Fabric. The F_Port shall respond by entering the OLS Receive State.

16.5.4.2 OLS Receive State (OL2)

A Port shall enter the OLS Receive State when it receives and recognizes the OLS Primitive Sequence. While in the OLS Receive State, the Port shall transmit the LR Primitive Sequence.

a) While in the OLS Receive State, the Port shall respond as follows:

– If LR is received and recognized, enter the LR Receive State (see 16.5.2.2).

– If LRR is received and recognized, enter the LRR Receive State (see 16.5.2.3).

– If NOS is received and recognized, enter the NOS Receive State (see 16.5.3.2).

– If OLS is received and recognized, remain in the OLS Receive State.

– If Idle is received and recognized, remain in the OLS Receive State.

b) If Loss of Synchronization is detected for more than a timeout period (R_T_TOV), or Loss of Signal is detected, the Port shall enter the Wait for OLS State (see 16.5.4.3). Detection of Loss of Signal or Loss of Synchronization shall not be counted as a Link Failure event in the Link Error Status Block.

16.5.4.3 Wait for OLS State (OL3)

A Port shall enter the Wait for OLS State when it detects Loss of Signal or Loss of Synchronization for more than a timeout period (R_T_TOV) while it is in the OLS Receive State, or while it is in the OLS Transmit State at an appropriate time in the Link Initialization Protocol (see 16.6.2).

Upon entry into the Wait for OLS State, the Port shall transmit the NOS Primitive Sequence. While in the Wait for OLS State, the Port shall respond as follows:

– If LR is received and recognized, enter the NOS Transmit State (see 16.5.3.1).

– If LRR is received and recognized, enter the NOS Transmit State (see 16.5.3.1).

– If NOS is received and recognized, enter the NOS Receive State (see 16.5.3.2).

— If OLS is received and recognized, enter the OLS Receive State (see 16.5.4.2).

— If Idle is received and recognized, remain in the Wait for OLS State.

16.6 Primitive Sequence Protocols

The interchange of frames is basically asynchronous in nature. The Data protocols defined in FC-PH allow multiple frames in transit in each direction at the same time. When conditions cause state changes to occur within an N_Port or F_Port, Primitive Sequence Protocols provide a fully interlocked and acknowledged mechanism to recover from those conditions.

See Q.3 and Q.4 for additional information describing Port state changes relative to Primitive Sequences. The Link recovery hierarchy is shown in figure 82 located in 29.4.

16.6.1 Primitive Sequence meanings

Table 27 - Primitive Sequence summary

Trans	Meaning	Resp
NOS	Not-Operational - Link Failure	OLS
OLS	Offline Sequence - Internal failure in Port - Transmitter may power-down, perform diagnostics, or perform initialization. - Receiver shall ignore Link errors or Link Failure.	LR
LR	Link Reset - Remove Class 1 Connection, - Reset F_Port, or - OLS recognized	LRR
LRR	Link Reset Response - Link Reset recognized	Idles
Idles	Operational Link	Idles

Table 27 provides a summary of the defined Primitive Sequences, their meaning, and the corresponding response when received.

16.6.2 Link Initializati n

Link initialization is required after a Port has been powered-on, has been internally reset, or has been Offline. The Ports involved may be an N_Port and an F_Port, or two N_Ports.

The following series of events defines the Link Initialization Protocol from the perspective of the Port which initiates the Protocol. While in the Offline State, NOS Reception or Link Failure conditions which are detected shall not be recorded as Link Failure events in the Link Error Status Block. The following series of events defines the Link Initialization Protocol.

a) Enter the OLS Transmit State and transmit OLS for a minimum period of 5 ms. After the period has elapsed, the Port shall respond as follows:

  — If LR is received and recognized, enter the LR Receive State (see 16.5.2.2).

  — If LRR is received and recognized, remain in the OLS Transmit State.

  — If OLS is received and recognized, enter the OLS Receive State (see 16.5.4.2).

  — If NOS is received and recognized, enter the NOS Receive State (see 16.5.3.2).

  — If Idle is received and recognized, remain in the OLS Transmit State.

  — If Loss of Synchronization is detected for more than a timeout period (R_T_TOV), or Loss of Signal is detected, enter the Wait for OLS State (see 16.5.4.3).

b) If no Primitive Sequence is received or event detected which causes the Port to exit the OLS Transmit State after a timeout period (R_T_TOV), the Port shall enter the Wait for OLS State.

At the end of the Link Initialization Protocol both Ports shall be transmitting and receiving Idles (Active State).

16.6.3 Onlin t Offlin

A Port is Online when it is in the Active State. A Port shall perform the Online to Offline Protocol to enter the Offline State. The Offline State shall be entered prior to power-down (or at power-down), or performing diagnostics. To exit the Offline State, a Port shall perform the Link Initialization Protocol.

The following series of events shall define the Online to Offline Protocol from the perspective of the Port which intends to go Offline.

a) The Port shall enter the OLS Transmit State.

b) The Port shall transmit OLS for a minimum time of 5 ms.

c) After transmitting OLS for 5 ms, the Port shall be Offline and may perform diagnostic procedures, turn off its transmitter, or transmit any signal (excluding Primitive Sequences) without errors being detected by the other attached Port. While in the Offline State, a Port may also power-down.

NOTE - After entering the OLS Transmit State and transmitting OLS for a minimum of 5 ms, the Port may then do anything which will not cause the attached Port to leave the OLS Receive State or Wait for OLS State unless intended to do so.

d) To exit the Offline State, a Port shall perform the Link Initialization Protocol.

16.6.4 Link Failure

The Link Failure Protocol shall be performed after a Port has detected either a Loss of Synchronization for a period of time greater than a timeout period (R_T_TOV), or Loss of Signal while not in the Offline State.

The Link Failure Protocol shall also be performed after a Link Reset Protocol timeout error is detected (see 16.6.5). The following series of events defines the Link Failure Protocol.

a) The Port shall transmit NOS and enter the NOS Transmit State while the Link Failure condition in the receiver or Port exists. While in the NOS Transmit State, after the Receiver is operational, the Port shall respond to Primitive Sequences received.

  — If NOS is received and recognized, enter the NOS Receive State.

  — If OLS is received and recognized, enter the OLS Receive State.

b) In the OLS Receive State, transmit LR.

  — If LR is received and recognized, enter the LR Receive State.

  — If LRR is received and recognized, enter the LRR Receive State.

c) In the NOS Receive State, transmit OLS.

  — If LR is received and recognized, enter the LR Receive State.

  — If OLS is received and recognized, enter the OLS Receive State.

d) In the LR Receive State, transmit LRR.

  — If Idles are received and recognized, exit the LR Receive State and enter the Active State.

e) In the LRR Receive State, transmit Idles.

  — If Idles are received and recognized, exit the LRR Receive State and enter the Active State.

f) At the end of the Link Failure Protocol both Ports shall be transmitting and receiving Idles (Active State).

16.6.5 Link Reset

The Link Reset Protocol shall be performed following a Link timeout. If a Port performs step a, b, or c without receiving an appropriate response within the timeout period (R_T_TOV), a Link Failure shall be detected and the Link Failure Protocol shall be performed.

The Link Reset Protocol shall comply with the following series of events described from the perspective of the Port initiating the Protocol.

a) The Port shall transmit LR and enter the LR Transmit State.

  — If LRR is received and recognized, enter the LRR Receive State.

  — If LR is received and recognized, enter the LR Receive State.

b) In the LR Receive State, transmit LRR.

— If Idles are received and recognized, exit the LR Receive State and enter the Active State.

c) In the LRR Receive State, transmit Idles.

— If Idles are received and recognized, exit the LRR Receive State and enter the Active State.

d) If other conditions or Primitive Sequences are recognized, the Port shall respond according to the definitions associated with its current State as specified in 16.4.

e) At the end of the Link Reset Protocol both Ports shall be transmitting and receiving Idles (Active State).

17 Frame formats

All FC-2 frames follow the general frame format as shown in figure 44. An FC-2 frame is composed of a Start_of_Frame delimiter, frame content, and an End_of_Frame delimiter. The frame content is composed of a Frame_Header, Data_Field, and CRC. Unless otherwise specified, the term frame refers to an FC-2 frame within FC-PH.

17.1 Frame transmission

Frame transmission shall be performed by inserting a frame immediately following a sequence of Idles. Idles shall be transmitted immediately upon completion of the frame.

At the N_Port transmitter there shall be a minimum of six Primitive Signals (Idles and R_RDY) between frames. A minimum of two Idles shall be guaranteed to precede the start (SOF) of each frame received by a destination N_Port. The Fabric may insert or remove Idles as long as the destination receives at least two Idles preceding each frame. See 17.8 for a discussion on frame reception.

17.2 Start_of_Frame (SOF)

The Start_of_Frame delimiter is an Ordered Set that immediately precedes the frame content. There are multiple **SOF** delimiters defined for the Fabric and for N_Port Sequence control. The **SOF** delimiter shall be transmitted on a word boundary. Tables 48 and 50 specify allowable delimiters by Class for Data and Link_Control frames. The bit encodings for the **SOF** delimiters are defined in table 24.

17.2.1 Start_of_Frame Connect Class 1 (SOF$_{c1}$)

SOF$_{c1}$ shall be used to request a Class 1 Dedicated Connection. The frame Data Field size is limited to the maximum size specified by the destination N_Port or by the Fabric, if present, whichever size is smaller. The maximum Data Field size is determined during the Login procedure. This delimiter also identifies the start of the first Sequence (i.e., implicit **SOF$_{i1}$**).

17.2.2 Start_of_Frame Initiate (SOF$_{ix}$)

A Sequence shall be initiated and identified by using the **SOF$_{ix}$** delimiter in the first frame. **SOF$_{ix}$** is used to indicate **SOF$_{i1}$**, **SOF$_{i2}$**, or **SOF$_{i3}$**. The following three delimiters identify the start of a Sequence based on Class of Service.

17.2.2.1 Start_of_Frame Initiate Class 1 (SOF$_{i1}$)

The first Sequence of a Dedicated Connection is initiated with **SOF$_{c1}$**. After a Class 1 Connection is established, all subsequent Sequences within that Dedicated Connection shall be initiated with **SOF$_{i1}$**.

17.2.2.2 Start_of_Frame Initiate Class 2 (SOF$_{i2}$)

The **SOF$_{i2}$** shall be used on the first frame to initiate a Sequence for Class 2 Service.

17.2.2.3 Start_of_Frame Initiate Class 3 (SOF$_{i3}$)

The **SOF$_{i3}$** shall be used on the first frame to initiate a Sequence for Class 3 Service.

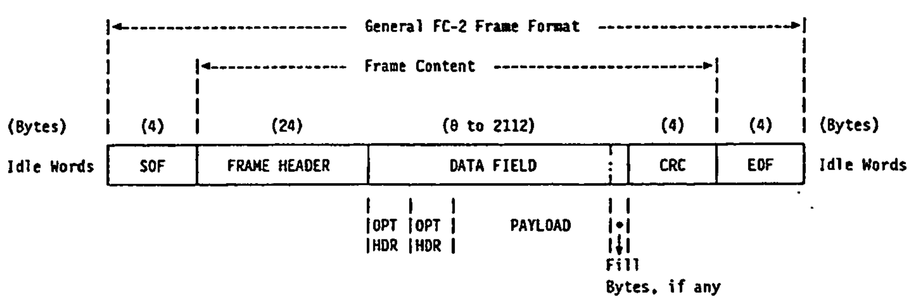


Figure 44 - FC-2 general frame format

17.2.3 Start_of_Frame Normal (SOFnx)

The following three delimiters identify the start of all frames other than the first frame of a Sequence based on Class of Service. **SOFnx** is used to indicate **SOFn1**, **SOFn2**, or **SOFn3**.

17.2.3.1 Start_of_Frame Normal Class 1 (SOFn1)

The **SOFn1** shall be used for all frames except the first frame of a Sequence for Class 1 Service.

17.2.3.2 Start_of_Frame Normal Class 2 (SOFn2)

The **SOFn2** shall be used for all frames except the first frame of a Sequence for Class 2 Service.

17.2.3.3 Start_of_Frame Normal Class 3 (SOFn3)

The **SOFn3** shall be used for all frames except the first frame of a Sequence for Class 3 Service.

17.2.4 Start_of_Frame Fabric (SOFf)

If a Fabric_Frame, indicated by **SOFf**, is received by an N_Port, the Fabric_Frame shall be discarded and ignored. See 17.9 for a description of a Fabric_Frame.

17.3 Frame_Header

The Frame_Header shall be the first field of the frame content and shall immediately follow the **SOF** for all frames. The Frame_Header shall be transmitted on a word boundary. The Frame_Header is used by the link control facility to control link operations, control device protocol transfers, and detect missing or out of order frames. The Frame_Header is described in detail in clause 18.

17.4 Data Field

The Data Field shall follow the Frame_Header. Data Fields shall be aligned on word boundaries and shall be equal to a multiple of four bytes in length (including zero length). Two Frame_Types are defined based on the value of bits 31-28 in the R_CTL field of the Frame_Header.

The two Frame_Types are

- FT_0 (Data Field = 0 bytes)

  When the R_CTL field bits 31-28 are set to 1 1 0 0, an FT_0 frame (Link_Control) shall be specified which has a Data Field length of zero.

- FT_1 (Data Field = 0 to 2 112 bytes)

  When the R_CTL field bits 31-28 are set to values other than 1 1 0 0, an FT_1 frame (Data frame) shall be specified whose Data Field size is a multiple of four bytes and ranges in size from 0 to 2 112 bytes. If the ULP supplies a Payload which is not divisible by 4, the remaining bytes (up to 3) shall contain fill bytes as specified in the F_CTL field in table 37. However, fill bytes shall only be meaningful, and recognized as present, on the last Data frame of a series of consecutive Data frames of a single Information Category within a single Sequence.

  The Data Field in FT_1 frames may contain optional headers, as defined by the DF_CTL field, as well as the Payload. Optional Data Field headers are described in clause 19.

A discussion of FT_0 (Link_Control) and FT_1 (Data) frames is found in clause 20.

17.5 CRC

The Cyclic Redundancy Check (CRC) is a four byte field that shall immediately follow the Data Field and shall be used to verify the data integrity of the Frame_Header and Data Field. **SOF** and **EOF** delimiters shall not be included in the CRC verification. The CRC field shall be calculated on the Frame_Header and Data Field prior to encoding for transmission and after decoding upon reception. The CRC field shall be aligned on a word boundary.

For the purpose of CRC computation, the bit of the word-aligned four byte field that corresponds to the first bit transmitted is the highest order bit. See figure N.1 in annex N. Bit 24 of the first word of the Frame_Header is the first bit of the transmission word transmitted by the transmitter (see 11.2.1).

ANSI X3.139, Fiber Distributed Data Interface - Media Access Control specifies the FC-PH CRC. See annex N for further discussions of the CRC. The CRC shall use the following 32-bit polynomial:

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$$

17.6 End_of_Frame (EOF)

The End_of_Frame (EOF) delimiter is an Ordered Set that immediately follows the CRC. The **EOF** delimiter shall designate the end of the frame content and shall be followed by Idles. Tables 48 and 50 specify allowable delimiters by Class for Data and and Link_Control frames. There are three categories of **EOF** delimiters. One category of delimiter shall indicate that the frame is valid from the sender's perspective and potentially valid from the receiver's perspective.

The second category shall indicate that the frame content is invalid. This category shall only be used by an F_Port which receives a complete frame and decodes it before forwarding that frame on to another destination.

The third category shall indicate the frame content is corrupted and the frame was truncated during transmission. The third category shall be used by both N_Ports and F_Ports to indicate an internal malfunction, such as a transmitter failure, which does not allow the entire frame to be transmitted normally.

The bit encodings for the **EOF** delimiters are defined in table 24.

17.6.1 Valid frame content

Three types of End_of_Frame delimiter are defined to indicate that the frame content is valid from the sender's perspective and potentially valid from the receiver's perspective. Tables 48 and 50 specify allowable delimiters by Class.

17.6.1.1 End_of_Frame Terminate (EOFt)

The **EOFt** shall indicat that the Sequ nce associated with this SEQ_ID is complete. **EOFt** or **EOFdt** shall be used to properly close a Sequence without error.

17.6.1.2 End_ f_Frame Disconn ct_Terminate (EOFdt)

The **EOFdt** shall remove a Dedicated Connection through a Fabric, if present. The **EOFdt** shall also identify the last ACK of a Sequence and indicate that all Class 1 Sequences associated with this S_ID are terminated. Open Class 1 Sequences, other than the SEQ_ID specified in the frame containing **EOFdt**, shall be abnormally terminated and may require Sequence recovery on a ULP protocol-dependent basis.

17.6.1.3 End_of_Frame Normal (EOFn)

The **EOFn** shall identify the end of frame when one of the other **EOF** delimiters indicating valid frame content is not required.

17.6.2 Invalid frame content

There are two **EOF** delimiters which indicate that the frame content is invalid. If a frame is received by a facility internal to a Fabric and an error is detected within the frame content, the frame may be passed to the destination N_Port with a modified **EOF** to indicate that an error was previously detected. Error detection in the Frame Content by the Fabric is optional.

Errors such as code violation or CRC errors are examples of detectable frame errors. For example, if the Fabric decodes a frame into bytes and one or more transmission characters cannot be validly decoded into bytes, these bytes shall be filled with valid data bytes, a new CRC shall not be generated, and the appropriate **EOF** delimiter shall replace the original **EOF** delimiter.

When a frame is received with an **EOF** delimiter which indicates that the frame content is invalid, the receiver shall not report the detection of an invalid frame. The invalid frame condition shall be reported by the entity which replaces the **EOF** delimiter that indicates invalid frame content. The receiving N_Port, at its discretion, may report the event of receiving a frame with one of these delimiters.

NOTES

1 Errors are counted at the point at which they are detected. Events may also be reported at the point at which they are recognized.

2 Due to the importance of the **EOFdt** delimiter in Class 1, it is very desirable for the Fabric to forward a frame which has either **EOFdt** or **EOFdti**.

The following **EOF** delimiters shall indicate that. the frame contents are invalid (see 17.8.2 regarding reception of invalid frames).

17.6.2.1 End_of_Frame Disconnect_Terminate_Invalid (EOFdti)

The **EOFdti** shall replace a recognized **EOFdt** delimiter on a frame with invalid frame content. The **EOFdti** shall remove a Dedicated Connection through a Fabric, if present. It shall also indicate that all Class 1 Open Sequences associated with the Connected N_Port are abnormally terminated and may require Sequence recovery on a ULP protocol-dependent basis.

The frame containing the **EOFdti** shall be processed by the receiver in the following manner:

- no response frame shall be transmitted,
- the Data Field may be used at the receiver's discretion (see 17.8.2), and
- the Dedicated Connection to the receiver is removed.

17.6.2.2 End_of_Frame Invalid (EOFni)

The **EOFni** shall replace an **EOFn** or **EOFt**, indicating that the frame content is invalid.

The frame containing the **EOFni** shall be processed by the receiver in the following manner:

- no response frame shall be transmitted, and
- the Data Field may be used at the receiver's discretion (see 17.8.2).

17.6.3 End_of_Frame Abort (EOFa)

The **EOFa** shall terminate a partial frame due to a malfunction in a link facility during transmission. The frame shall end on a word boundary and shall be discarded by the receiver without transmitting a reply. The transmitter shall retransmit the aborted frame with the same sequence count (SEQ_CNT) up to its ability to retry, which may b zero.

An invalid **EOF** (**EOFdti** or **EOFni**) delimiter may be changed to an **EOFa** delimiter under the conditions specified for **EOFa**.

EOFa delimiters shall not be changed to an invalid **EOF** delimiter under any conditions.

17.7 Frame field order

The frame content shall be transmitted sequentially following the **SOF** delimiter as an ordered byte stream within the definition of the frame as specified in figures 44, 46, and 47 until the **EOF** delimiter is transmitted.

Figure 45 relates the frame format to the ordered byte stream transferred from the Upper Level Protocol (or FC-4) and transmitted by FC-PH.

Word	Bits 3322 2222 1098 7654	2222 1111 3210 9876	1111 11 5432 1098	7654 3210
SOF	K28.5 A0	Dxx.x A1	Dxx.x A2	Dxx.x A3
0	R_CTL B0	B1	D_ID B2	B3
1	rrrr rrrr B4	B5	S_ID B6	B7
2	TYPE B8	B9	F_CTL B10	B11
3	SEQ_ID B12	DF_CTL B13	SEQ_CNT B14	B15
4	OX_ID B16	B17	RX_ID B18	B19
5	B20	Parameter B21	B22	B23
6	B24	Payload B25	B26	B27
7	B28	Payload B29	B30	B31
8	B32	CRC B33	B34	B35
EOF	K28.5 C0	Dxx.x C1	Dxx.x C2	Dxx.x C3

Figure 45 - Frame Byte Order

A frame shall be transmitted in the following byte order:
A0, A1, A2, A3, B0, B1, B2, B3, B4 ...
B32, B33, B34, B35, C0, C1, C2, C3.

If there were one byte of fill in the Payload of this frame, the fill byte is B31. With no optional header present, the Relative Offset (Parameter Field) shall be specified as follows:
Relative Offset + 0 specifies B24
Relative Offset + 3 specifies B27
Relative Offset + 4 specifies B28

For a Relative Offset of decimal 1024 (hex '00000400') the Parameter Field shall be specified as:
B20, B21, B22, B23 = hex '00 00 04 00'.

See figure N.1 and N.2 in annex N.

17.8 Frame reception

The following list specifies frame reception rules.

a) Data bytes received outside the scope of a delimiter pair (**SOF** and **EOF**) shall be discarded.

b) Frame reception shall be started by recognition of an **SOF** delimiter.

c) Recognition of **SOFc1**, **SOFx2**, or **SOFx3**, shall be responded to by transmission of the R_RDY Primitive Signal when the interface buffer is available.

d) Detection of a code violation after frame reception is started but before frame reception is terminated shall be identified as an Invalid Transmission Word within the frame.

e) Frame reception shall continue until an Ordered Set, or a Link Failure is detected.

f) If the Frame_Content between the **SOF** and **EOF** delimiters exceeds the maximum allowable Data Field size as specified by Login, an N_Port may consider the frame invalid and discard Data Field bytes as received. However, frame reception shall still be terminated by an Ordered Set or Link Failure. An N_Port is also allowed to receive the frame and if valid, it shall respond with a P_RJT with a length error indicated in the reason code.

g) In either process or discard policy, if frame reception is terminated by an **EOFa**, the entire frame shall be discarded, including the Frame_Header and Data Field.

h) If an **EOFdt** or **EOFdti** terminates frame reception, a pending or existing Dedicated Connection shall be recognized to be removed without regard to frame validity.

17.8.1 Frame validity

The following list of items determines whether a frame is considered valid or invalid.

a) If the Ordered Set terminating frame reception is not recognized as an **EOF** delimiter, the frame shall be considered invalid.

b) If the **EOF** delimiter is **EOFni**, **EOFdti**, or **EOFa**, the frame shall be considered invalid.

c) If the Frame_Content between the **SOF** and **EOF** delimiters is not a multiple of four bytes, the frame shall be considered invalid.

d) If the **EOF** delimiter is **EOFn**, **EOFt**, or **EOFdt** and no Invalid Transmission Words have been detected during frame reception for a frame with a multiple of four byte words, the CRC field shall be used. If the CRC is valid, the frame shall be considered valid and normal processing for valid frames shall be performed. If the CRC is invalid, the frame shall be considered invalid.

During normal processing of valid frames, errors may be detected which are rejectable in Class 1 and 2 using the P_RJT Link_Response frame (see 20.3.3.3). P_RJT frames shall not be transmitted for invalid frames. If a rejectable error condition or a busy condition is detected for a valid Class 3 frame, the frame shall be discarded.

When errors such as Invalid Transmission Word and Invalid CRC are detected, the event count in the Link Error Status Block shall be updated (see 29.8). If delimiter usage does not follow allowable delimiters by class as specified in tables 48 and 50, a valid frame shall be considered rejectable.

17.8.2 Invalid frame processing

If an N_Port is able to determine that an invalid frame is associated with an Exchange which is designated as operating under Process policy, th N_Port may process and use th Data Field at its discretion, otherwise, the entire invalid frame shall be discarded.

NOTE - When a frame is corrupted, it is not known if the Frame_Header is correct. The XIDs, SEQ_ID, SEQ_CNT, and Parameter fields may not contain reliable information. The error may cause a misrouted frame to have a D_ID which appears to be correct. Such a frame may be used under very restricted conditions.

17.9 Fabric_Frame

A Fabric_Frame may be used by the Fabric for intrafabric communication. A Fabric_Frame is composed of an **SOFf** delimiter, frame header and content, and an **EOFn** delimiter. The Fabric_Frame header and content is not defined within FC-PH.

18 Frame_Head r

18.1 Introduction

The Frame_Header shall be subdivided into fields as shown in figure 46.

Word	Bits 3322 2222 1098 7654	2222 1111 3210 9876	1111 11 5432 1098	7654 3210
0	R_CTL	D_ID		
1	rrrr rrrr	S_ID		
2	TYPE	F_CTL		
3	SEQ_ID	DF_CTL	SEQ_CNT	
4	OX_ID		RX_ID	
5	Parameter			

Figure 46 - Frame_Header

The Frame_Header shall be the first field of the frame content and immediately follow the **SOF** delimiter. The Frame_Header shall be transmitted on a word boundary. The Frame_Header is used to control link operations, control device protocol transfers, and detect missing or out of order frames.

18.1.1 Frame identification

A single frame is uniquely identified by S_ID, D_ID, SEQ_ID, SEQ_CNT, and Sequence Context. The OX_ID and RX_ID fields (generally referred to as X_ID) may be used by an N_Port to provide a locally assigned value which may be used in place of S_ID, D_ID, and SEQ_ID to identify frames in a non-streamed Sequence or when only one Sequence is Open. When Sequences are streamed, or more than one Sequence is Open, the X_ID field may be used in place of the S_ID and D_ID to identify the N_Port pair associated with a specific frame. The X_ID field may also be used in conjunction with S_ID, D_ID, or SEQ_ID to relate one or more Sequences to actions initiated by Upper Level Protocols.

When one or more Sequences are aborted using the Abort Sequence Protocol (see 29.7.1.1), a Recovery_Qualifier range is identified by the Sequence Recipient which consists of S_ID, D_ID, OX_ID, RX_ID in combination with a range of SEQ_CNT values (low and high). In Class 2 and 3, the Recovery_Qualifier range shall be used by the Sequence Initiator to discard ACK and Link_Response frames and by the Sequence Recipient to discard Data frames. If a Recovery_Qualifier is used in Class 2 or 3, a Reinstate Recovery Qualifier Link Service request shall be performed after an R_A_TOV timeout period has expired (see 21.2.2).

18.1.2 Sequence identification

The set of IDs described previously (S_ID, D_ID, OX_ID, RX_ID, and SEQ_ID) is referred to as the Sequence_Qualifier throughout the remainder of FC-PH. An N_Port implementation makes use of these IDs in an implementation-dependent manner to uniquely identify Active and Open Sequences (see 24.6.1).

NOTE - An N_Port's freedom to assign a SEQ_ID is based on Sequence Context (Initiator or Recipient). This may affect the means by which an N_Port implementation chooses to uniquely identify Sequences.

A given N_Port Originator may choose to provide frame tracking outside of the signaling protocol of FC-PH (FC-2). This is indicated by setting the OX_ID to hex 'FFFF'. This implies that the Exchange Originator shall only have one Exchange Active with a given destination N_Port. If an N_Port chooses an alternative frame tracking mechanism outside the scope of FC-2, it is still responsible for providing proper SEQ_ID and SEQ_CNT values. In addition, it shall return the RX_ID assigned by the Responder.

A given N_Port Responder may choose to provide frame tracking outside of the signaling protocol of FC-PH (FC-2). This is indicated by setting the RX_ID to hex 'FFFF'. If an N_Port chooses an alternative frame tracking mechanism outside the scope of FC-2, it is still responsibl for providing proper SEQ_ID and SEQ_CNT values. In addition, it shall return the OX_ID assigned by the Originator.

18.2 R_CTL fi ld

Routing Control (Word 0, Bits 31-24) is a one byte field that contains routing bits and information bits to categorize the frame function.

When the R_CTL field is used in combination with the TYPE field (Word 2, bits 31-24), it provides an N_Port with assistance in frame routing, data routing, or addressing as summarized in table 28.

- **Bits 31-28 - Routing Bits**

  0000 = FC-4 Device_Data frame
  0010 = Extended Link_Data frame
  0011 = FC-4 Link_Data frame
  0100 = Video_Data frame
  1000 = Basic Link_Data frame
  1100 = Link_Control frame
  Others = Reserved

Routing bits differentiate frames based on function or service within an N_Port or F_Port. FC-4 Device_Data frames contain Payload information related to a specific Upper Level Protocol. Video_Data frames contain Payload information directed to a video buffer without transferring the Data to main store.

Link_Data frames contain request and reply commands which are directed to the N_Port. There are three kinds of Link_Data frames:

— Basic Link_Data which contain commands in the information bits of R_CTL (bits 27-24) which are part of a Sequence of frames,
— Extended Link_Data frames which are directed to the N_Port in order to provide N_Port Extended Link Services, such as Login, which are common to multiple FC-4s, and
— FC-4 Link_Data frames which are directed to the N_Port in order to provide FC-4 Services to assist in the processing of FC-4 Device_Data frames.

See clause 20 for a specification of Data (FC-4 Device_Data and Video_Data) and Link_Control frames. See clause 21 for a specification on Basic Link Services, Extended Link Services, and FC-4 Link Services.

- **Bits 27-24 - Information field**

The interpretation of the Information field is dependent on the Routing (bits 31-28) field value. For Routing = 1000 and 1100, the Information field contains a command. For all other R_CTL values, the Information field specifies the Common Information Categories specified in table 29. When the Routing bits indicate a Link_Control frame, the command code is specified in the Information field (bits 27-24) as shown in table 49. Link_Control frames are described in clause 20.

Table 28 - R_CTL - TYPE CODE SUMMARY

R_CTL Wd 0, bits 31-24		TYPE Wd 2, bits 31-24	Payload	Comments
Bits 31-28 Routing	Bits 27-24 Information			
0000	0000-0111 (see table 29)	FC-4 Device_Data (see table 36)	Information Categories 0 to 15	Device_Data frames
0010	0010 0011 (see table 29)	Extended Link_Data (see table 34)	Request Reply	Extended Link Service
0011	0010 0011 (see table 29)	FC-4 Link_Data (see table 36)	Request Reply	FC-4 Link_Data
0100	0000-0111 (see table 29)	Video_Data (see table 35)		Video_Data
1000	Command (see table 57)	Basic Link_Data (see table 34)	12 byte max	Basic Link Service
1100	Command (see table 49)	Reserved / Reason Code	none	Link_Control frame

When the Routing bits indicate a Basic Link_Data frame, the command code shall be specified in the Information field (bits 27-24) as shown in table 57.

When the Routing bits indicate Extended Link_Data or FC-4 Link_Data, the Information Category shall indicate Unsolicited Control for a request Sequence and Solicited Control for a reply Sequence as encoded in table 29.

When the Routing bits indicate FC-4 Device_Data or Video_Data, the Information Category bits are available to each FC-4 TYPE to indicate data type or data control information relating to the information within the Payload portion of the Data Field as shown in table 29. When the Routing bits indicate Device_Data, it indicates that the Payload of this frame may be processed by a level above the N_Port which received the frame. The current TYPEs for FC-4 Device_Data and FC-4 Link_Data frames are shown in table 36.

Table 29 - Information Categories

Bit value 27-24	Description
0000	Uncategorized information
0001	Solicited Data
0010	Unsolicited Control
0011	Solicited Control
0100	Unsolicited Data
0101	Data Descriptor
0110	Unsolicited Command
0111	Command Status
Others	Unspecified

Category bits shall specify an Information Category for the Payload of a single frame. A series of consecutive frames which specify the same Information Category within a single Sequence shall be considered a single instance of the Information Category. The fill bits (F_CTL bits 1-0) shall only be meaningful on the last frame of a single instance of an Information Category. Only one instance of an Information Category shall be allowed in a single Sequence. Multiple Information Categories are allowed in the same Sequence as a Login option (see 23.6.8.4).

The format of the Payload for Data_Descriptor category is shown in table 30. The format of the Payload for Unsolicited Command category is shown in table 31. The format of the Payload for Command Status is shown in table 32. The format of the Payload of other categories are FC-4 specific.

Table 30 - Data Descriptor Payload

Item	Size -Bytes
Offset of Data being transferred	4
Length of Data being transferred	4
Reserved	4
Other optional information (FC-4 dependent)	max

Table 31 - Unsolicited Command Payload

Item	Size -Bytes
Entity Address (FC-4 dependent)	8
Command information (FC-4 dependent)	max

Table 32 - Command Status Payload

Item	Size -Bytes
Command status (bits 31 - 0) Bit 31 = 0 Successful = 1 Unsuccessful Bit 30 = 0 Complete = 1 Incomplete	4
Reserved	4
Optional status (FC-4 dependent)	max

18.3 Address identifiers

Each N_Port shall have a native N_Port Identifier which is unique within the address domain of a Fabric. In addition, an N_Port may optionally have one or more alias address identifiers which may be shared across multiple N_Ports. For example, alias addressing may be used to implement a Hunt Group.

An N_Port Identifier of binary zeros indicates that an N_Port is unidentified. When an N_Port transitions from the Offline State to the Online

State, its N_Port Identifier shall be unidentified. While an N_Port is unidentified, it shall

- promiscuously accept frames with any Destination Identifier value,
- not Reject (P_RJT) a frame with a reason code of Invalid D_ID, and
- Reject (P_RJT) frames other than Basic and Extended Link Service with a reason code of Login Required.

An N_Port determines its native N_Port address Identifier by performing the Login Procedure as specified in clause 23. During the Login Procedure an N_Port may be assigned a native N_Port Identifier or it may determine its own native N_Port Identifier.

Address identifiers in the range of hex 'FFFFF0' to 'FFFFFE' are well-known and reserved for use as shown in table 33. The address identifier of hex 'FFFFFF' is reserved as a broadcast address.

Table 33 - Well-known address identifiers

Address Value	Description
FFFFF0 to FFFFF9	Reserved
FFFFFA	Management Server
FFFFFB	Time Server
FFFFFC	Directory Server
FFFFFD	Fabric Controller
FFFFFE	Fabric F_Port

18.3.1 Destination_ID (D_ID)

The Destination Identifier (D_ID) is a three byte field (Word 0, Bits 23-0) that shall contain the address identifier of an N_Port or F_Port within the destination entity.

18.3.2 Source_ID (S_ID)

The Source Identifier (S_ID) is a three byte field (Word 1, Bits 23-0) that shall contain the address identifier of an N_Port or F_Port within the source entity.

18.4 Data structur type (TYPE)

The data structure type (TYPE) is a one byte field (Word 2, Bits 31-24) that shall identify the protocol of the frame content for Data frames. The TYPE field specified identifies the TYPE at the Sequence Recipient for Data frames.

When the Routing bits in R_CTL indicate a Link_Control frame other than F_BSY, the TYPE field is reserved. F_BSY frames use the TYPE field to indicate a reason code for the F_BSY in TYPE field bits 31-28. When the F_BSY is in response to a Link_Control frame, R_CTL bits 27-24 of the busied frame are copied by the Fabric into the TYPE bits 27-24. The bit encodings are shown in table 49. Duplication of the Link_Control command code allows an N_Port to easily retransmit the frame if it is busied by the Fabric (see 20.3.3.1).

When the Routing bits in R_CTL indicate Basic or Extended Link_Data, TYPE codes are decoded as shown in table 34.

Table 34 - TYPE codes - N_Port/F_Port Link Service

Encoded Value Wd 2, bits 31-24	Description
0000 0000	Basic Link Service
0000 0001	Extended Link Service
0000 0010 to 1100 1111	Reserved
1101 0000 to 1111 1111	Vendor Unique

When the Routing bits in R_CTL indicate Video_Data, the TYPE codes are decoded as shown in table 35.

Table 35 - TYPE codes - Video_Data

Encoded Value Wd 2, bits 31-24	Description
0000 0000 to 1100 1111	Reserved
1101 0000 to 1111 1111	Vendor Unique

Table 36 - TYPE codes - FC-4 (D vice_Data and Link_Data)

Encoded Value Wd 2, bits 31-24	Description
0000 0000	Reserved
0000 0001	Reserved
0000 0010	Reserved
0000 0011	Reserved
0000 0100	ISO/IEC 8802 - 2 LLC (In order)
0000 0101	ISO/IEC 8802-2 LLC/SNAP
0000 0110	Reserved
0000 0111	Reserved
0000 1000	SCSI - FCP
0000 1001	SCSI - GPP
0000 1010 to 0000 1111	Reserved - SCSI
0001 0000	Reserved - IPI-3
0001 0001	IPI-3 Master
0001 0010	IPI-3 Slave
0001 0011	IPI-3 Peer
0001 0100	Reserved for IPI-3
0001 0101	CP IPI-3 Master
0001 0110	CP IPI-3 Slave
0001 0111	CP IPI-3 Peer
0001 1000	Reserved - SBCCS
0001 1001	SBCCS - Channel
0001 1010	SBCCS - Control Unit
0001 1011 to 0001 1111	Reserved - SBCCS
0010 0000	Fibre Channel Services
0010 0001	FC-FG
0010 0010	FC-XS
0010 0011	FC-AL
0010 0100	SNMP
0010 0101 to 0010 0111	Reserved - Fabric Services
0010 1000 to 0010 1111	Reserved - Futurebus
0100 0000	HIPPI - FP
0100 0001 to 0100 0111	Reserved - HIPPI
0100 1000 to 1101 1111	Reserved
1110 0000 to 1111 1111	Vendor Unique

When the Routing bits in R_CTL indicate FC-4 Device_Data or FC-4 Link_Data TYPE codes are decoded as shown in table 36.

18.5 Frame Control (F_CTL)

The Frame Control (F_CTL) field (Word 2, Bits 23-0) is a three byte field that contains control information relating to the frame content. The following subclause describes the valid uses of F_CTL bits. If an error in bit usage is detected, a reject frame (P_RJT) shall be transmitted in response with an appropriate reason code (see 20.3.3.3) for Class 1 and 2. The format of the F_CTL field is defined in table 37.

Bit 23 - Exchange Context

An Exchange shall be started by the Originator facility within an N_Port. The destination N_Port of the Exchange shall be known as the Responder. Each frame for this Exchange indicates whether the S_ID is associated with the Originator (0) or Responder (1).

Bit 22 - Sequence Context

A Sequence shall be started by a Sequence Initiator facility within an N_Port. The destination N_Port of the Sequence shall be known as the Sequence Recipient. Each frame of the Sequence indicates whether the S_ID is associated with the Sequence Initiator (0) or the Sequence Recipient (1).

NOTE - Ownership is required for proper handling of Link_Control frames received in response to Data frame transmission in Class 2. When a Busy frame is received, it may be in response to a Data frame (Sequence Initiator) or to an ACK frame (Sequence Recipient). This bit simplifies the necessary constructs to distinguish between the two cases.

Bit 21 - First_Sequence

This bit shall be set to one on all frames in the First Sequenc of an Exchange. It shall be set to zero for all other Sequences within an Exchange.

Tabl 37 (Pag 1 of 2) - F_CTL field			
Control Field	**Word 2, Bits**	**Description**	**Refer-ence**
Exchange/Sequence Control	23-14		
Exchange Context	23	0 = Originator of Exchange 1 = Responder of Exchange	see 24.4
Sequence Context	22	0 = Sequence Initiator 1 = Sequence Recipient	see 24.4
First_Sequence	21	0 = Sequence other than first of Exchange 1 = first Sequence of Exchange	see 24.5
Last_Sequence	20	0 = Sequence other than last of Exchange 1 = last Sequence of Exchange	see 24.7
End_Sequence	19	0 = Data frame other than last of Sequence 1 = last Data frame of Sequence	see 24.6.3
End_Connection	18	0 = Connection active 1 = End of Connection Pending (Class 1)	see 28.7.2
Chained_Sequence	17	0 = No Chained_Sequence 1 = Chained_Sequence Active (Class 1)	see 24.6.5
Sequence Initiative	16	0 = hold Sequence Initiative 1 = transfer Sequence Initiative	see 24.6.3
X_ID reassigned	15	0 = X_ID assignment retained 1 = X_ID reassigned	see 25.3.2
Invalidate X_ID	14	0 = X_ID assignment retained 1 = invalidate X_ID	see 25.3.1
Reserved	13-10		
Retransmited Sequence	9	0 = Original Sequence transmission 1 = Sequence retransmission	see 29.7.1.2
Unidirectional Transmit	8	0 = Bidirectional transmission 1 = Unidirectional transmission	see 28.5.3
Continue Sequence Condition	7-6	Last Data frame - Sequence Initiator 0 0 = No information 0 1 = Sequence to follow-immediately 1 0 = Sequence to follow-soon 1 1 = Sequence to follow-delayed	see 24.6.5
Abort Sequence Condition	5-4	ACK frame - Sequence Recipient 0 0 = Continue Sequence 0 1 = Abort Sequence, Perform ABTS 1 0 = Stop Sequence 1 1 = Immediate Sequence retransmission requested Data frame (1st of Exchange) - Sequence Initiator 0 0 = Abort, Discard multiple Sequences 0 1 = Abort, Discard a single Sequence 1 0 = Process policy with infinite buffers 1 1 = Discard multiple Sequences with immediate retransmission	see 24.6.5 and 21.2.2
Relative Offset present	3	0 = Parameter field not meaningful 1 = Parameter Field = Relative Offset	
Exchange reassembly	2	Reserved for Exchange reassembly	

Table 37 (Page 2 of 2) - F_CTL field			
Control Field	**Word 2, Bits**	**Description**	**Ref r-ence**
Fill Data Bytes	1-0	End of Data field - bytes of fill 0 0 = 0 Bytes of fill 0 1 = 1 Byte of fill (last byte of Data field) 1 0 = 2 Bytes of fill (last 2 bytes of Data field) 1 1 = 3 Bytes of fill (last 3 bytes of Data field)	

Bit 20 - Last_Sequence

This bit shall be set to one on the last Data frame in the Last Sequence of an Exchange. This bit is permitted to be set to one on a Data frame prior to the last frame. Once it is set to one, it shall be set to one on all subsequent Data frames in the last Sequence of an Exchange. It shall be set to zero for all other Sequences within an Exchange. This bit shall be set to the same value in the Link_Control frame as the Data frame to which it corresponds.

NOTE - The early transition of this bit, unlike other F_CTL bits, is permitted as a hardware assist by providing an advance indication that the Sequence is nearing completion.

Bit 19 - End_Sequence

This bit shall be set to one on the last Data frame of a Sequence. In Class 1, if this bit is set to one in the ACK corresponding to the last Data frame, it confirms that the Sequence Recipient recognized it as the last Data frame of the Sequence. In Class 2, the final ACK with this bit set to one confirms the end of the Sequence, however, the SEQ_CNT shall match the last Data frame delivered, which may not be the last Data frame transmitted. This indication is used for Sequence termination by the two N_Ports involved in the Sequence in addition to **EOFt** or **EOFdt** (see 24.3.8). This bit shall be set to zero for other frames within a Sequence.

Bit 18 - End_Connection (E_C)

The E_C bit shall be set to one in the last Data frame of a Sequence to indicate that the N_Port transmitting E_C is beginning the disconnect procedure. The N_Port transmitting E_C set to one on the last Data frame of a Sequence is requesting the receiving N_Port to transmit an ACK frame terminated by **EOFdt** if the receiving N_Port has completed all Active Sequences. If the receiving N_Port is not able to transmit **EOFdt**, E_C set to one requests that the receiving N_Port complete all Active Sequences and not initiate any new Sequences during the current Connection.

The E_C bit is only applicable to Class 1 Service and is only meaningful on the last Data frame of a Sequence. The E_C bit shall be set to zero on a connect-request frame (**SOFc1**) in order to avoid ambiguous error scenarios where the ACK (**EOFdt**) is not properly returned to the Connection Initiator (see 28.7.2).

Receiving the C_S bit set to one, overrides any previous transmission of the E_C bit set to one. See 28.7.2 for a discussion on removing Dedicated Connections (E_C bit).

Bit 17 - Chained_Sequence (C_S)

The Chained_Sequence bit shall be set to one on the last Data frame of a Sequence to indicate that the Sequence Initiator requires a reply Sequence from the Sequence Recipient within the existing Dedicated Connection. This bit is only meaningful on a Data frame when the End_Sequence bit is set to one, Sequence Initiative bit set to one, and Unidirectional Transmit bit set to zero. When the Sequence Recipient receives the C_S bit set to one, it shall respond to the request by initiating a reply Sequence even if it had previously transmitted the E_C bit set to one. This bit is only applicable to Class 1 Service.

NOTE - The C_S bit is provided to support existing system architectures which require a chained function such as command or status transfer.

Bit 16 - Sequence Initiative

The Originator of an Exchange shall initiate the first Sequence as the Sequence Initiator. If the Sequence Initiative bit is set to zero, the Sequence Initiator shall hold the initiative to continue transmitting Sequences for the duration of this Sequence Initiative. Th Sequence Recipient gains the initiative to transmit a new Sequence for this Exchange after the Sequence

Initiative has been transferred to the Recipient. This shall be accomplished by setting the Sequence Initiative bit to one in the last Data frame of a Sequence (End_Sequence = 1). In Class 1 and 2, the Sequence Initiator shall consider Sequence Initiative transferred when the ACK to the corresponding Data frame is received with the Sequence Initiative bit = 1. Setting bit 16 = 1 is only meaningful when End_Sequence = 1.

Bit 15 - X_ID reassigned

Bit 15 is only meaningful if an N_Port requires or supports X_ID reassignment as specified during N_Port Login. See clause 25, 25.3.2, and 25.5 for specification on bit 15 usage. Otherwise, bit 15 is not meaningful.

Bit 14 - Invalidate X_ID

Bit 14 is only meaningful if an N_Port requires or supports X_ID reassignment as specified during N_Port Login. See clause 25, 25.3.1, and 25.5 for specification on bit 14 usage. Otherwise, bit 14 is not meaningful.

Bit 9 - Retransmitted Sequence

Bit 9 is only meaningful in Class 1 and only if the Exchange Error Policy specified on the first Data frame of the Exchange by the Originator is specified as 1 1 (Discard multiple Sequences with immediate retransmission). Bit 9 shall be set = 1 if the Sequence Initiator has received an ACK with the Abort Sequence Condition bits (F_CTL bits 5-4) set = 1 1. Bit 9 shall be set = 1 in all frames of the retransmitted Sequence. If the Sequence Initiator is not able to determine that all Sequences prior to the Sequence identified in the ACK with bits 5-4 set = 1 1 (i.e., missing final ACK) are complete, the Sequence Initiator shall not retransmit any Sequences until the successful reception of all previous Sequences has been verified using a Read Exchange Status (RES) Extended Link Service request (see 29.7.1.2) or other such method.

Bit 8 - Unidirectional Transmit

Bit 8 is meaningful on a connect-request (**SOFc1**) in Class 1. If bit 8 is set = 0, the Dedicated Connection is bidirectional. If a Dedicated Connection is bidirectional, the Connection Recipient may initiate Sequences immediately after the Dedicated Conn ction is established. If bit 8 is s t = 1, the Dedicated Connection is unidirectional and only the N_Port which transmitted the connect-request which establishes the Connection shall transmit Data frames.

Other than the connect-request, bit 8 is meaningful on the first and last Data frames of a Sequence. After the connect-request with bit 8 set = 1, the Connection Initiator may reset bit 8 = 0 making the Connection bidirectional for the duration of the Connection on the first or last Data frames of a Sequence (i.e., the bit is not meaningful in other Data frames of the Sequence). The Connection Recipient may request that a unidirectional Connection be changed to a bidirectional Connection by setting bit 8 = 0, in an ACK frame. Once set to zero in an ACK, all subsequent ACKs shall be transmitted with bit 8 = 0. The Connection Initiator is not required to honor the request to become bidirectional. See 28.5.3 for additional clarification.

Bits 7-6 - Continue Sequence Condition

The Continue Sequence Condition bits are information bits which may be set to indicate an estimated transmission time for the next consecutive Sequence of the current Exchange. The Continue Sequence Condition bits are meaningful on a Data frame when the End_Sequence bit is set to one and the Sequence Initiative bit is set to zero. The Continue Sequence Condition bits are not meaningful on a Data frame when the End_Sequence bit is set to one and the Sequence Initiative bit is set to one.

The Continue Sequence Condition bits are also meaningful on the ACK to the last Data frame (End_Sequence = 1) when Sequence Initiative is also transferred (Sequence Initiative bit = 1). In this case, the Continue Sequence Condition bits indicate how long it will be until the N_Port which received Sequence Initiative will transmit its first Sequence for this Exchange.

The Continue Sequence Condition bits may be used to manage link resources within an N_Port such as X_ID reassignment and maintaining or removing an existing Class 1 Connection. The bits are informational and may be used to improve the performance and management of link resources within an N_Port. The bits are not binding. The tim estimate is relative to the time to remove and reestablish a Class 1 Connection regardless of the Class of Service being us d.

Sequence Condition bits:

```
0 0 = No information
0 1 = Sequence to follow - immediately
1 0 = Sequence to follow - soon
1 1 = Sequence to follow - delayed
```

When the Continue Sequence Condition bits are set to 0 0, no information is being offered regarding when the next Sequence is being transmitted.

When the Continue Sequence Condition bits are set to 0 1, this indicates that the next consecutive Sequence for this Exchange shall be transmitted immediately.

When the Continue Sequence Condition bits are set to 1 0, this indicates that the next consecutive Sequence for this Exchange shall be transmitted in a time period which is less than the time to remove and reestablish a Class 1 Connection (soon).

When the Continue Sequence Condition bits are set to 1 1, this indicates that the next consecutive Sequence transmission for this Exchange shall be delayed for a period of time which is longer than the time to remove and reestablish a Class 1 Connection. If the Sequence Initiator holds Sequence Initiative and indicates delayed transmission of the next Sequence, the Initiator shall wait until the final ACK (**EOFt**) before transmitting the next Sequence for the Exchange.

Bits 5-4 - Abort Sequence Condition

The Abort Sequence Condition bits shall be set to a value by the Sequence Initiator on the first Data frame of an Exchange to indicate that the Originator is requiring a specific error policy for this Exchange. The Abort Sequence Condition bits shall not be meaningful on other Data frames within an Exchange. The error policy passed in the first frame of the first Sequence of an Exchange shall be the error policy supported by both N_Ports participating in the Exchange (see 29.6.1.1).

```
0 0 = Abort, Discard multiple Sequences
0 1 = Abort, Discard a single Sequence
1 0 = Process policy with infinite buffers
1 1 = Discard multiple Sequences with
        immediate retransmission
```

In the Abort, Discard multiple Sequences Error Policy, the Sequence Recipient shall deliver Sequences to the FC-4 or upper level in the order transmitted under the condition that the previous Sequence, if any, was also deliverable. If a Sequence is determined to be non-deliverable, all subsequent Sequences shall be discarded until the ABTS protocol has been completed. The Abort, Discard multiple Sequences Error Policy shall be supported in Class 1, 2, or 3.

In the Abort, Discard a single Sequence Error Policy, the Sequence Recipient may deliver Sequences to the FC-4 or upper level in the order that received Sequences are completed by the Sequence Recipient without regard to the deliverability of any previous Sequences. The Abort, Discard a single Sequence Error Policy shall be supported in Class 1, 2, or 3.

In the Process policy with infinite buffers, frames shall be delivered to the FC-4 or upper level in the order transmitted. Process policy with infinite buffers shall use ACK_0 (see 20.3.2.2) and shall only be allowed in Class 1.

In the Discard multiple Sequences with immediate retransmission Error Policy, the Sequence Recipient shall deliver Sequences to the FC-4 or upper level in the order transmitted under the condition that the previous Sequence, if any, was also deliverable. If a Sequence is determined to be non-deliverable, all subsequent Sequences shall be discarded until a new Sequence is received with the Retransmission bit (Bit 9) in F_CTL set = 1 or until the ABTS protocol has been completed. The Discard multiple Sequences with immediate retransmission is a special case of the Discard multiple Sequences Error Policy. This policy is applicable to an Exchange where all transmission is in Class 1.

Process policy support shall be indicated by an N_Port during N_Port Login. Discard policy shall be supported.

NOTE - If the delivery order of Sequences, without gaps, is required by an FC-4 to match the transmission order of Sequences within an Exchange, then one of the two Discard multiple Sequence Error Policies is required. In the Discard a Single Sequence Error Policy, out of order Sequence delivery is to be expected and handled by the FC-4 or upper level.

The Abort Sequence Condition bits shall be set to a value other than zeros by the Sequence Recipient in an ACK frame to indicate to the Sequence Initiator that an abnormal condition, malfunction, or error has been detected by the Sequence Recipient.

```
0 0 = Continue Sequence
0 1 = Abort Sequence requested
1 0 = Stop Sequence
1 1 = Immediate Sequence retransmission
        requested
```

A setting of 0 1 indicates a request by the Sequence Recipient to the Sequence Initiator to terminate this Sequence using the Abort Sequence protocol and then optionally perform Sequence recovery. See 21.2.2 and 29.7.1.1 for a description of the Abort Sequence protocol.

A setting of 1 0 indicates a request by the Sequence Recipient to the Sequence Initiator to stop this Sequence. This allows for a request for an early termination by the Sequence Recipient. Some of the data received may have been processed and some of the data discarded. Aborting the Sequence using the ABTS command is not necessary and shall not be used. Both the Sequence Initiator and Recipient end the Sequence in a normal manner. See 29.7.2 for a description of the Stop Sequence protocol.

A setting of 1 1 indicates that the Sequence Recipient has detected an error in a Class 1 Sequence and requests that the Sequence Initiator begin immediate retransmission of the Sequence, if able (see 29.7.1.2). Sequence retransmission also uses F_CTL bit 9. The Sequence status is saved by the Sequence Recipient in the Exchange Status Block associated with the aborted SEQ_ID.

Bit 3 - Relative Offset present

When bit 3 is set to zero on a Data frame, the Parameter Field is not meaningful. When bit 3 is set to one on a Data frame, the Parameter Field contains the Relative Offset for the Payload of the frame. Bit 3 is only meaningful on Data frames of a Sequence and shall be ignored on ACK and Link_Response frames. Bit 3 is not meaningful on Basic Link_Data frames.

Bit 2 - reserved for Exchange reassembly

The Sequence Initiator shall set bit 2 to 0 to indicate that the Payload in this Data frame is associated with an Exchange between a single pair of N_Ports. Therefore, reassembly is confined to a single destination N_Port.

NOTE - Bit 2 being set to 1 is reserved for future use to indicate that the Payload in this Data frame is associated with an Exchange being managed by a single Node using multiple N_Ports at either the source, destination, or both.

Bits 1-0 - Fill Data Bytes

When the bits associated with the Fill Data Bytes are non-zero, it notifies the Data frame receiver (Sequence Recipient) that one or more of the last three bytes of the Data Field shall be ignored, except for CRC calculation. The fill byte value is not specified by FC-PH but shall be a valid data byte.

Bits 1-0 shall only be meaningful on the last Data frame of a series of consecutive Data frames of a single Information Category within a single Sequence. For example, if a Sequence contains Data frames of a single Information Category, non-zero values for bits 1-0 shall only be meaningful on the last Data frame of the Sequence. The fill Data bytes shall not be included in the Payload.

18.5.1 F_CTL Summary

[7] Tables 38 and 39 are provided to summarize the relationship of F_CTL bits on Data frames and on Link_Response frames. The text describing each F_CTL bit more clearly elaborates on the manner in which F_CTL bits interact with each other. The tables provide an overall summary.

Bits 15 and 14 are only meaningful if an N_Port requires or supports X_ID reassignment as specified during N_Port Login. The bits are included in the table for completeness.

When a bit is designated as meaningful under a set of conditions, that bit shall be ignored if those conditions are not present. For example, bit 18 is only meaningful when bit 19 = 1; this means that bit 18 shall be ignored unless bit 19 = 1.

[7] The future enhancement to the Fibre Channel architecture will address the control bit needed for high speed hardware implementation of multiple categories in a Sequence.

18.5.1.1 F_CTL bits on Data frames

Table 38 shows the key interactions between specific bits within the F_CTL field. The top part of table 38 describes those bits which are unconditionally meaningful on the first, last, or any Data frame of a Sequence. The connect-request may reflect settings for either the first Data frame of a Sequence, the last Data frame of a Sequence, or both first and last.

NOTE - A key control function may become effective when a F_CTL bit is set to 1. The locations where the key function is meaningful are indicated in the top part of the table 38.

The bottom part of table 38 describes those bits which are conditionally meaningful. For example, Bit 19 = 1 (vertical column) is only meaningful on the last Data frame of a Sequence. Bit 18 = 1 (vertical column) is only meaningful on the last Data frame when bit 19 = 1. Bit 17 (vertical column) is only meaningful on the last Data frame when bit 19 = 1, bit 18 = 0, and bit 16 = 1.

Table 38 - F_CTL bit interactions on Data frames

Bits associated with Data frame order:	23	22	21 =1	20 =1	19 =1	18 =1	17 =1	16 =1	15	14	9 =1	8 =1	7-6	5-4	3 =1	1-0
1st frame of Seq	M	M	M						M+		M	M		MF	M	M
last frame of Seq	M	M	M		M				M+		M	M			M	M
any frame of Seq	M	M	M						M+		M				M	M
connect-request	M	M	M	ML	ML		ML		M+ ML	ML		M	ML		M	M
Bits 20-0 (above) are meaningful on Data frames when - the bit in the Left-hand column is:																
First_Sequence																
21=0																
21=1														MF		
Last_Sequence																
20=0																
20=1																
End_Sequence																
19=0																
19=1				ML		ML	ML	ML		ML			ML			
End_Connection																
18=0							ML									
18=1																
Chained_Sequence																
17=0																
17=1																
Sequence Initiative																
16=0																
16=1							ML									

NOTES
1 M = Meaningful
2 MF = Only meaningful on First Data frame of a Sequence
3 ML = Only meaningful on Last Data frame of a Sequence
4 M+ = Meaningful on first and following Data frames of a Sequence until at least one ACK is received for Sequence (X_ID reassign)

18.5.1.2 F_CTL bits n Link_Control

Table 39 - F_CTL bit interactions on ACK, BSY, or RJT

Bits associated with ACK frame order:	23	22	21	20	19	18	17	16	15	14	9 =1	8 =1	7-6	5-4	3	1-0
ACK to 1st frame	V	V	E						M+		M	M		Ma		
ACK to last frame	V	V	E						M+		M	M	ML	Ma		
ACK to any frame	V	V	E						M+		M	M		Ma		
ACK to connect-request	V	V	E	E	ML		ML	ML	M+ ML	ML		M	ML	Ma		
The Bits below are meaningful on the ACK, BSY, or RJT for the corresponding Data frame																
Exchange Context																
23=0	V															
23=1	V															
Sequence Context																
22=0		V														
22=1		V														
First_Sequence																
21=0			E													
21=1			E													
Last_Sequence																
20=0				E												
20=1				E												
End_Sequence																
19=0					E											
19=1					ML	ML	ML	ML		ML			ML			
End_Connection																
18=0						E										
18=1						ML										
Chained_Sequence																
17=0							E									
17=1							M									
Sequence Initiative																
16=0								E								
16=1							ML	ML								

NOTES
1 M = Meaningful
2 Ma = Meaningful on ACK frames
3 ML= Only meaningful on Last ACK, BSY, RJT frame of a Sequence
4 E = Echo (meaningful)
5 V = Inverse or invert (meaningful)
6 M+ = Meaningful on First and following ACK frames until Sequence Recipient X_ID appears in a Data frame (X_ID reassignment)

Table 39 shows the key interactions with F_CTL bits on ACK, BSY, and RJT frames and should be reviewed together with table 38. F_CTL bits 19, 18, 17, and 16 in an ACK frame are retransmitted to reflect confirmation (1) or denial (0) of those indications by the Sequence Recipient. For example, if bits 5-4 are set to (0 1) in response to a Data frame in which bit 19 = 1 and bit 16 = 1, setting bits 19 and 16 to zero in the ACK frame indicates that the Data frame was not processed as the last Data frame and that Sequence Initiative was not accepted by the Sequence Recipient of the Data frame since the Sequence Recipient is requesting that the

Sequence Initiator transmit an ABTS frame to Abort the Sequence. See 24.3.8, 24.3.10, and 29.7.1.1 for additional information on setting the Abort Sequence Condition bits.

The ACK to a connect-request may reflect settings for either the first Data frame of a Sequence, the last Data frame of a Sequence, or both first and last.

NOTE - A key control function may become effective when a F_CTL bit is set to 1. The locations where the key function is meaningful are indicated in the top part of the table 39.

18.6 Sequence_ID (SEQ_ID)

The SEQ_ID is a one byte field (Word 3, Bits 31-24) assigned by the Sequence Initiator which shall be unique for a specific D_ID and S_ID pair while the Sequence is Open. Both the Sequence Initiator and the Sequence Recipient track the status of frames within the Sequence using fields within the Sequence_Qualifier. If its X_ID is unassigned, it shall use any other field or fields such as S_ID, D_ID, or the other N_Port's X_ID for tracking (see 18.1.1 and 18.9).

If the Sequence Initiator initiates a new Sequence for the same Exchange before receiving the final ACK (**EOFt**, **EOFdt**) for the previous Sequence in Class 1 and 2, or before R_A_TOV has expired for all frames of a Class 3 Sequence, it is termed a streamed Sequence. If streamed Sequences occur, it is the responsibility of the Sequence Initiator to use X+1 different consecutive SEQ_IDs where X is the number of Open Sequences per Exchange (see 23.6.8.8). For example, if X=2 from Login, then consecutive SEQ_IDs of 11-93-22-11-93 is acceptable.

If consecutive non-streamed Sequences for the same Exchange occur during a single Sequence Initiative, it is the responsibility of the Sequence Initiator to use a different SEQ_ID for each consecutive Sequence. For example, consecutive SEQ_IDs of 21-74-21-74 is acceptable. The examples show when a SEQ_ID shall be allowed to be repeated. A series of SEQ_IDs for the same Exchange may also be random and never repeat (also see 24.3.4). See 24.6.2 for more discussion regarding reusing and timing out Recovery_Qualifiers following an aborted or abnormally terminated Sequence, or an aborted Exchange.

The combination of Initiator and Recipient Sequence Status Blocks identified by a single SEQ_ID describe the status of that Sequence for a given Exchange. See 24.8.2 for a description of the Sequence Status Block maintained by the Sequence Recipient.

18.7 DF_CTL

Data_Field Control (DF_CTL) is a one byte field (Word 3, Bits 23-16) that specifies the presence of optional headers at the beginning of the Data_Field for Device_Data or Video_Data frames. DF_CTL bits are not meaningful on Link_Control or Basic Link Service frames. DF_CTL bit 22 is the only meaningful DF_CTL bit on Extended or FC-4 Link Service frames. Control bit usage is shown in table 40.

Table 40 - DF_CTL bit definition

Word 3, Bit(s)	Optional Header
23	Reserved for Extended Frame_Header
22	0 = No Expiration_Security Header 1 = Expiration_Security Header
21	0 = No Network_Header 1 = Network_Header
20	0 = No Association_Header 1 = Association_Header
19-18	Reserved
17-16	0 0 = No Device_Header 0 1 = 16 Byte Device_Header 1 0 = 32 Byte Device_Header 1 1 = 64 Byte Device_Header

The Optional Headers shall be positioned in the Data Field in the order specified with the bit 23 header as the first header in the Data Field, bit 22 header as the second header in the Data Field, and so forth, in a left to right manner corresponding to bits 23, 22, 21, and so forth as shown in figure 47.

If either bit 17 or 16 is set to one, then a Device Header is present. The size of the Device Header is specified by the encoded value of bits 17 and 16 as shown.

If an Optional Header is not present as indicated by the appropriate bit in DF_CTL, no space shall

be allocated for the Header in the Data Field of the frame. Therefore, for example, if bits 23 and 22 are zero and bit 21 is one, the first data byte of the Data Field contains the first byte of the Network_Header.

See clause 19 for a discussion on Optional Headers.

18.8 Sequence count (SEQ_CNT)

The sequence count (SEQ_CNT) is a two byte field (Word 3, Bits 15-0) that shall indicate the sequential order of Data frame transmission within a single Sequence or multiple consecutive Sequences for the same Exchange. The sequence count of the first Data frame of the first Sequence of the Exchange transmitted by either the Originator or Responder shall be binary zero. The sequence count of each subsequent Data frame in the Sequence shall be incremented by one.

If a Sequence is streamed, the sequence count of the first Data frame of the Sequence shall be incremented by one from the sequence count of the last Data frame of the previous Sequence (this is termed continuously increasing SEQ_CNT). If a Sequence is non-streamed, the starting sequence count may be continuously increasing or binary zero.

ACK and Link_Response frames shall be identified by the same SEQ_ID and SEQ_CNT as the frame to which it is responding. Frames are tracked on a SEQ_ID, SEQ_CNT basis within the scope of the Sequence_Qualifier for that Sequence.

The sequence count shall wrap to zero after reaching a value of 65 535. The sequence count shall then only be incremented to (but not including) the sequence count of an unacknowledged frame of the same Sequence. Otherwise, data integrity is not ensured. Sequences of Data frames and sequence count values are discussed in clause 24. See 23.6.8.7 regarding data integrity, Credit, and sequence count.

18.9 Originat r Exchange_ID (OX_ID)

The Originator Exchange_ID is a two byte field (Word 4, Bits 31-16) that shall identify the Exchange_ID assigned by the Originator of the Exchange. Each Exchange shall be assigned an identifier unique to the Originator or Originator-Responder pair. If the Originator is enforcing uniqueness via the OX_ID mechanism, it shall assign a unique value for OX_ID other than hex 'FFFF' in the first Data frame of the first Sequence of an Exchange. An OX_ID of hex 'FFFF' indicates that the OX_ID is unassigned and that the Originator is not enforcing uniqueness via the OX_ID mechanism. If an Originator uses the unassigned value of hex 'FFFF' to identify the Exchange, it shall have only one Exchange (OX_ID= hex 'FFFF') with a given Responder.

An Originator Exchange Status Block associated with the OX_ID is used to track the progress of a series of Sequences which comprises an Exchange. See clause 24 for a discussion of Sequences and Exchanges. See 24.8.1 for a description of the Exchange Status Block.

NOTE - If hex 'FFFF' is used as the OX_ID throughout the Exchange, then the Originator uses an alternate Sequence tracking mechanism. If the OX_ID is unique, it may be used as an index into a control block structure which may be used in conjunction with other constructs to track frames.

18.10 Responder Exchange_ID (RX_ID)

The Responder Exchange_ID is a two byte field (Word 4, Bits 15-0) assigned by the Responder which shall provide a unique, locally meaningful identifier at the Responder for an Exchange established by an Originator and identified by an OX_ID. The Responder of the Exchange shall assign a unique value for RX_ID other than hex 'FFFF', if RX_ID is being used, in an ACK to a Data frame in the first Sequence of an Exchange in Class 1 and 2, or in the first Sequence transmitted as a Sequence Initiator, if any, in Class 3. An RX_ID of hex 'FFFF' shall indicate that the RX_ID is unassigned. If the Responder does not assign an RX_ID other than hex 'FFFF' by the end of the first Sequence, then the Responder is not enforcing uniqueness via the RX_ID mechanism.

When the Responder uses only hex 'FFFF' for RX_ID, it shall have the capability to identify the Exchange through the OX_ID and the S_ID of the Originator of the Exchange. Under all other circumstances, until a value other than hex 'FFFF' is assigned, hex 'FFFF' value for RX_ID shall be used indicating that RX_ID is unassigned. After a value other than hex 'FFFF' is assigned, the assigned value shall be used for the remainder of the Exchange (see 24.3.2 item e and 24.5.2).

A Responder Exchange Status Block associated with the RX_ID is used to track the progress of a series of Sequences which compose an Exchange. See 24.8.1 for a description of the Exchange Status Block.

See clause 24 for a discussion of Sequences and Exchanges.

NOTE - If hex 'FFFF' is used as the RX_ID throughout the Exchange, then the Responder uses an alternate Sequence tracking mechanism. If the RX_ID is unique, it may be used as an index into a control block structure which may be used in conjunction with other constructs to track frames.

18.11 Parameter

The Parameter field (Word 5, Bits 31-0) has two meanings based on frame type. For Link_Control frames, the Parameter field is used to carry information specific to the individual Link_Control frame. For Data frames, the Parameter field specifies Relative Offset, a four-byte field that contains the relative displacement of the first byte of the Payload of the frame from the base address as specified by the ULP. Relative Offset is expressed in terms of bytes (see 17.7).

The use of the Relative Offset field is optional and is indicated as a Login Service Parameter. The setting of F_CTL bit 3 determines whether the Parameter Field shall be meaningful as a Relative Offset for Data frames.

The offset value shall be relative to an Information Category within a Sequence for an Exchange. If Relative Offset is being used, the number of bytes transmitted in a single Sequence shall not exceed the maximum value of the Relative Offset (Parameter) field (2^{32}).

NOTE - Performance may be improved if data is aligned on natural boundaries.

See clause 27 for a discussion concerning Relative Offset. See clause 20 for a discussion concerning use of the Parameter field in Link_Control frames.

19 Optional headers

19.1 Introduction

Optional headers defined within the Data Field of a frame are

a) Expiration_Security_Header
b) Network_Header
c) Association_Header
d) Device_Header.

The presence of optional headers is defined by control bits in the DF_CTL field of the Frame_Header. The sequential order of the optional headers, Payload, and their sizes are indicated in figure 47.

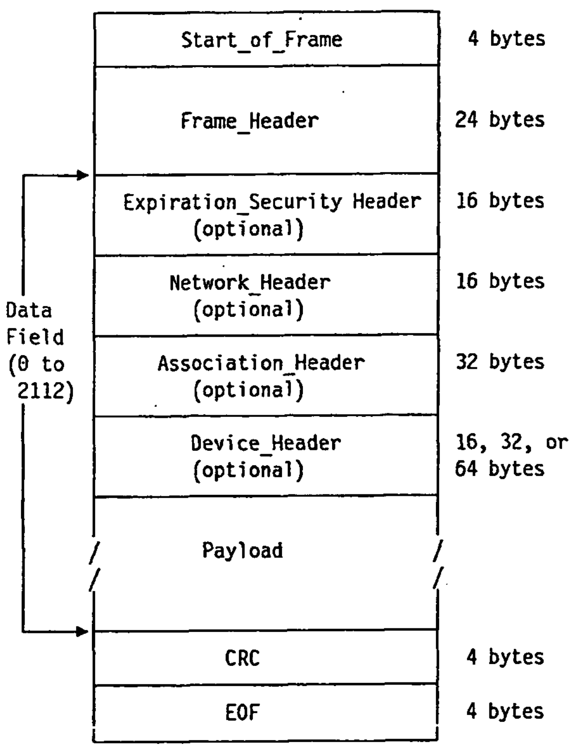


Figure 47 - Optional headers order

If present, an Expiration_Security_Header shall be the first optional header to follow the Frame_Header. If present, a Network_Header shall be the next 16 bytes of the Data Field. If present, an Association_Header shall be the next 32 bytes of the Data Field. If present, a Device_Header shall follow the optional headers. If none of the optional headers is present, no space in the Data Field shall be reserved.

19.2 Expiration_Security_Header

The Expiration_Security_Header, as shown in figure 48, is an optional header within the Data_Field content. Its presence shall be indicated by bit 22 in the DF_CTL field, located in the Frame_Header, being set to one. The Expiration_Security_Header shall be 16 bytes in size.

19.2.1 Expiration Time

An Expiration Timer in a system shall be used to compute the Expiration Time to be included in this field. The Expiration Timer shall be a 64-bit fixed-point number in seconds. The integer part of the number shall be the most significant 32 bits, and the fractional part shall be the least significant 32 bits. The maximum integer value for the expiration time is $2^{32}-1$ s with 32 bits for the fraction of a second.

On start up, a value of zeros shall be used to indicate an invalid or undefined time. When an N_Port begins communication within a system, it may obtain the Expiration Timer value from the well-known Time Server which may be physically resident in the Fabric or other specified N_Port. The Expiration Time shall be determined by adding an expiration time value to the current system Expiration Timer value. If a frame is received after the Expiration Time has been exceeded, the frame shall be discarded by the N_Port. The frame may be discarded by the Fabric without notification.

If multiple Expiration Timers are present in the Fabric, the common start time relative to 0000 Universal Time (UT) on 1 January 1900 shall be used. These timers shall be synchronized to an accuracy of $\pm$ 2 s. With a single Expiration Timer present in the Fabric, an alternate time base may be used for the start time. The fractional part for this timer is optional.

19.2.2 R lationship to R_A_TOV

If the Expiration Time value is such that the R_A_TOV value (see 29.2.1), is exceeded prior to reaching the Expiration Time for a frame, the rules regarding R_A_TOV shall be followed regardless of the value of the Expiration Time. If the Expiration Time is reached prior to exceeding R_A_TOV, the N_Port, upon discarding the frame, should account for it in such a way as to avoid a Sequence Timeout.

NOTE - If a Fabric discards a frame which has reached its Expiration Time prior to exceeding R_A_TOV, a Sequence Timeout will result which could otherwise be avoided.

19.2.3 Security type (S_Type)

The field indicates the type of Security supported.

19.2.4 Security length (S_Length)

The field indicates the length in bytes of the security information.

19.2.5 Security

If the security information indicated by S_Length is ≤ 4, then this field shall contain the security information.

If the security information indicated by S_Length is > 4, then this field is not meaningful and shall not contain the security information. The Payload shall start with the security information. The security information may be followed by other user data in the Payload.

Word ↓ / Bits →			
0	63 Expiration Time (Most significant word) 32		
1	31 Expiration Time (Least significant word) 0		
2	31 24 S_Type	Reserved	15 S_Length 0
3	31 Security 0		

Figure 48 - Expiration_Security_Header

19.3 Network_Header

The Network_Header may be used by a bridge or a gateway node which interfaces to an external Network. The Network_Header, if present, shall be 16 bytes in size.

The Network_Header, as shown in figure 49, is an optional header within the Data_Field content. Its presence shall be indicated by bit 21 in the DF_CTL field, located in the Frame_Header, being set to one. The Network_Header may be used for routing between Fibre Channel networks of different Fabric address spaces, or Fibre Channel and non-Fibre Channel networks. The Network_Header contains Name_Identifiers for Network_Destination_Address and Network_Source_Address. The Name_Identifiers permitted in the Network_Header are shown in table 42.

Bits →
Word ↓

Word		
0	63 60 D_NAA	59 ... 32 Network_Destination_Address (high order bits)
1	31 ... 0 Network_Destination_Address (low order bits)	
2	63 60 S_NAA	59 ... 32 Network_Source_Address (high order bits)
3	31 ... 0 Network_Source_Address (low order bits)	

Figure 49 - Network_Header

19.3.1 D_NAA or S_NAA

Destination Network_Address_Authority (D_NAA) or Source Network_Address_Authority (S_NAA) field indicates the authority responsible for the administration of the network address (destination or source) used. The Network_Address_Authority indicators are shown in table 41.

Table 41 - NAA identifiers

Bits 63 62 61 60	NAA
0 0 0 0	ignored
0 0 0 1	IEEE
0 0 1 0	IEEE extended
0 0 1 1	Locally assigned
0 1 0 0	IP
0 1 0 1	Reserved
...	...
1 0 1 1	Reserved
1 1 0 0	CCITT - individual address
1 1 0 1	Reserved
1 1 1 0	CCITT - group address
1 1 1 1	Reserved

19.3.2 Network_Destination_ID or Netw rk_S urce_ID

The Network_Destination_ID or Network_Source_ID shall be a 60 bit field indicating the network address being used.

19.3.2.1 IEEE 48-bit address

When D_NAA (or S_NAA) is IEEE, Network_Destination_ID (or Network_Source_ID) field shall contain a 48-bit IEEE Standard 802.1A Universal LAN MAC Address (ULA). The ULA shall be represented as an ordered string of six bytes numbered from 0 to 5. The least significant bit of byte 0 shall be the Individual/Group Address (I/G) bit. The next least significant bit shall be the Universally or Locally Administered Address (U/L) bit. IEEE Standard 802.1A further specifies that the bytes be transferred in the order 0 to 5. Figure 50 shows how the bytes of an ULA shall be mapped to two words on the Network Header.

Users
A unique 48 bit IEEE address may be assigned to an N_Port, a Node, an F_Port, or a Fabric.

Bits	63	55	47	39	32
	0 0 0 1	Zeros	ULA Byte 0 U/L I/G		ULA Byte 1
	ULA Byte 2	ULA Byte 3	ULA Byte 4		ULA Byte 5
Bits	31	23	15	7	0

Figure 50 - IEEE 48-bit address format

19.3.2.2 IEEE extended

When D_NAA (or S_NAA) is IEEE extended, Network_Destination_ID (or Network_Source_ID) field shall contain the 48-bit IEEE address assigned to a single field replaceable hardware unit, preceded by a 12 bit address uniquely indicating an F_Port or an N_Port contained in that unit.

Users
A unique IEEE extended address may be assigned to an F_Port or an N_Port.

19.3.2.3 Locally assigned

When D_NAA (or S_NAA) is locally assigned, Network_Destination_ID (or Network_Source_ID) shall be assigned by the local environment and shall be Fabric unique.

Users
A Fabric unique address may be locally assigned to an N_Port, a Node, an F_Port, or a Fabric.

19.3.2.4 32-bit IP address

When D_NAA (or S_NAA) is IP, Network_Destination_ID (or Network_Source_ID) field shall contain a 32-bit IP address. Figure 51 shows how the bytes of an IP address shall be mapped to two words on the Network Header.

Users
A unique 32 bit IP address may be assigned to a Node. Assignment of IP addresses shall conform to accepted Internet Protocol conventions.

NOTE - It is reasonable to have a D_NAA indicate an IP address and S_NAA indicate that the Network_Source_ID is ignored.

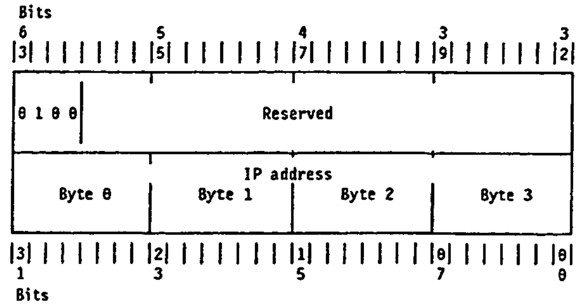


Figure 51 - 32-bit IP address format

19.3.2.5 CCITT 60-bit address

When D_NAA (or S_NAA) is CCITT - individual or group address, Network_Destination_ID (or Network_Source_ID) field shall contain a 60-bit CCITT individual or group address respectively.

Users
A unique 60 bit CCITT individual address or a 60 bit CCITT group address may be assigned to an N_Port, a Node, an F_Port, or a Fabric.

19.3.3 Application summary

The application of Name_Identifiers in Network_Header for heterogeneous (FC to Non-FC) networks and homogeneous (FC to FC) networks is summarized in table 42.

Table 42 - Network addresses

NAA	Name_Identifier	Network
IEEE	WWN	Heterogeneous
CCITT - individual address	WWN	Heterogeneous
CCITT - group address	WWN	Heterogeneous
IP	WWN	Heterogeneous
IEEE extended	FCN	FC Networks
Local	FCN	FC Networks

Note:
WWN - Worldwide Name (worldwide unique address)
FCN - Fibre Channel Name (Fibre Channel unique address)

FC Name_Identifier

The Network_Adresses used in the Network_Header may refer to various FC entities. The Name_Identifiers used for various FC entities are summarized in table 43.

NOTE - FC-PH does not prevent a Fabric Element from being assigned a unique Worldwide Name in addition to those of its F_Ports. However such usage is outside the scope of FC-PH.

Name_Identifier formats

Formats for various Name_Identifiers are summarized in table 44.

Table 43 - Fibre Channel user identifiers

NAA	Name_Identifier	Fibre Channel users			
		N_Port	Node	F_Port	Fabric
IEEE	WWN	yes	yes	yes	yes
CCITT - individual address	WWN	yes	yes	yes	yes
CCITT - group address	WWN	yes	yes	yes	yes
IEEE extended	FCN	yes	no	yes	no
Local	FCN	yes	yes	yes	yes

Note: yes - applicable to the user
no - not applicable to the user
WWN - Worldwide Name (worldwide unique identifier)
FCN - Fibre Channel Name (Fibre Channel unique identifier)

Table 44 - Name_Identifier formats

NAA	Name_Identifier (64 bits)		
	NAA ID (4 bits)	60 bit field	
		(12 bits)	(48 bits)
IEEE	0001	zeros	IEEE address
IEEE extended	0010	N_Port identifier within the Node	IEEE address for Node
		F_Port identifier within the Fabric element	IEEE address for Fabric element
Local	0011	Fabric unique	
IP	0100	zeroes (28 bits)	IP address (32 bits)
CCITT - individual address	1100	CCITT address	
CCITT - group address	1110	CCITT address	

19.4 Ass ciation_Header

The Association_Header is an optional header within the Data_Field content. Its presence shall be indicated by bit 20 in the DF_CTL field, located in the Frame_Header, being set to one. The Association_Header shall be 32 bytes in size.

The Association_Header may be used to locate an Exchange Status Block when an X_ID is invalidated during an Exchange (see 25.3.1 and 25.3.2). The Association_Header may also be used to identify a specific Process or group of Processes within a Node associated with an Exchange. When an N_Port has indicated during Login that an Initial Process_Associator is required to communicate with it, the Association_Header shall be used by that N_Port to identify a specific Process or group of Processes within a Node associated with an Exchange (see 25.1). The Association_Header shall be used for either one or both of these two functions.

The Association_Header shall be subdivided into fields as illustrated in figure 52. The Validity bits (V) shall indicate whether each of the four Associators contain meaningful (valid) information (bit = 1), or that the Associator shall be ignored (bit = 0) as defined in table 45. The contents of each Associator of the Association_Header are meaningful to the Node which generates that particular field. They may not be meaningful to the other Node receiving it.

Applicability
Association_Header is applicable to Class 1, 2, or 3. The use of the Association_Header for X_ID Reassignment shall be restricted to Class 1 and 2.

NOTE - The Association_Header is provided to support system architectures which require more than two levels of identifiers, i.e., X_ID and SEQ_ID. For an example of four level identifier usage, see annex R.

Word	Bits →		
0	31	vvvv rrrr rrrr rrrr rrrr rrrr rrrr rrrr	0
1	31	Originator Process_Associator	0
2	31	rrrr rrrr rrrr rrrr rrrr rrrr rrrr rrrr	0
3	31	Responder Process_Associator	0
4	63	Originator Operation_Associator (most significant word)	32
5	31	Originator Operation_Associator (least significant word)	0
6	63	Responder Operation_Associator (most significant word)	32
7	31	Responder Operation Associator (least significant word)	0

Figure 52 - Association_Header

Table 45 specifies the meaning of the Validity bits (V) in figure 52.

Table 45 - Association_Header Validity bits (Word 0, Bits 31 - 28)

Bit	Description
31	Originator Process_Associator 0 = not meaningful 1 = meaningful
30	Responder Process_Associator 0 = not meaningful 1 = meaningful
29	Originator Operation_Associator 0 = not meaningful 1 = meaningful
28	Responder Operation_Associator 0 = not meaningful 1 = meaningful

19.4.1 Pr cess_Associators

Process_Associators (Originator and Responder) have the following characteristics:

- A Process_Associator identifies a specific process or group of processes within a Node.
- The Process_Associator is the mechanism by which a specific process or group of processes is addressed by another communicating process.

NOTE - An example of a group of processes is a set of related processes controlled by a single instance of an operating system.

- A Process_Associator shall be specified by the Node owning the specific process or group of processes and shall be meaningful to that Node. The value specified may not be meaningful to the other communicating Nodes, although the value may have been made known to them. The other communicating Nodes shall return the Process_Associators given to them.
- A Process_Associator, once assigned for an Exchange, shall not be changed within the life of the Exchange.
- A Process_Associator shall not be required to be remembered after the logout of the communicating N_Port.
- The contents of Process_Associator fields are implementation dependent.
- A Process_Associator shall span, within the Node, all N_Ports having access to the related process.

19.4.2 Operation_Associators

Operation_Associators (Originator and Responder) have the following characteristics:

- An Operation_Associator identifies an operation which is a Node specific construct within a given Node.
- Operation_Associator is the mechanism by which this Node specific construct is referred to by another communicating Node.
- An Operation_Associator shall be specified by the Node owning the operation and shall be meaningful only to that Node. The value specified may not be meaningful to the other communicating Node, although the value may hav been made known to them. The value assigned to an Op ration_Associator is made available to other communicating Node through transmission of an Association_Header (see 25.1). The mechanism by which a value is assigned to an Operation_Associator is not specified in FC-PH.
- Operation_Associators shall be remembered for the life of an operation.
- Operation_Associators shall not change for the life of an operation, including the spanning of disconnects in Class 1.
- The contents of Operation_Associator fields are implementation dependent.
- An Operation_Associator may span all N_Ports having access to the related operation within a Node.

19.5 Device_Header

The Device_Header, if present, shall be 16, 32, or 64 bytes in size. The contents of the Device Header are entirely under the control of a level above FC-2 based on the TYPE field.

19.6 Optional header usage

19.6.1 Expiration_Security_Header

Expiration_Security_Header, if used, shall be present either in the first frame or in all frames of a Sequence. If the receiving N_Port does not support the header function, it shall reject the header with the reject reason code of Security_Expiration_Header not supported.

19.6.2 Network_Header

Network_Header, if used, shall be present only in the first Data frame of a Sequence. If the receiving N_Port does not support the header function, it shall ignore the header and skip the Data field by the header length (16 bytes).

19.6.3 Association_Header

Association_Header if present, shall be present only in the first Data frame of a Sequ nce (see 25.2).

19.6.4 Devic _Head r

Device_Header, if present, shall be present only in the first Data frame of a Sequence. A Device_Header may be used by a ULP type. For that ULP type, the Device_Header is required to be supported. The Device_Header may be ignored and skipped, if not needed. If a Device_Header is present for a ULP which does not require it, the related FC-4 may reject the frame with the reason code of TYPE not supported.

19.6.5 Link Servic

No Optional_Headers are permitted in Basic Link Service. Only the Expiration_Security_Header is permitted for use with Extended Link Service and FC-4 Link Service.

19.6.6 Summary

Table 46 summarizes usage of Optional headers.

Table 46 - Optional header usage summary

Optional header	Where present	Sequence applicability	Receiving N_Port action
Expiration_Security _Header	Either in the first Data frame or all Data frames of a Sequence	All Sequences except Basic Link Services	Rejects if not supported
Network_Header	Only in the first Data frame of a Sequence	All Sequences except Basic and Extended Link Services	Skips if not supported or required
Association_Header	Only in the first Data frame of a Sequence	Only in Sequences defined in Association_ Header management protocols (see 25.2)	See 25.2 and 25.3
Device_Header	Only in the first Data frame of a Sequence	All Sequences except Basic and Extended Link Services	Skips if not needed or FC-4 may reject if the related ULP Type does not support it

20 Data frames and respons s

20.1 Introduction

Table 47 identifies the types of frames that are possible to be transmitted and received within an N_Port.

20.2 Data frames

As shown in table 47, two types of frames are defined:

- Data frames, and
- Link_Control frames.

Data frames defined include:

- Link_Data,
- Device_Data, and
- Video_Data.

When the term "Data frame" is used in FC-PH, it refers to any of the types of Data frames that may be transmitted. The rules of transmission of Data frames and response frames (Link_Control) are equally applicable to each type of Data frame.

Data frames may be used to transfer information such as data, control information, header information, and status from a source N_Port to a destination N_Port. In Class 1 and 2, each Data frame successfully transmitted shall be acknowledged to indicate successful delivery to the destination N_Port. An indication of unsuccessful delivery of a valid frame shall be returned to the transmitter by a Link_Response frame in Class 1 and 2.

Data frames may be streamed, i.e., multiple frames may be transmitted by a single N_Port before a response frame is received. The number of outstanding, unacknowledged Data frames allowed is specified by a Class Service Parameter during the Login procedure (end-to-end Credit). See clause 23 for the specification of Login and Service Parameters and clause 26 for the specification of flow control rules.

Table 47 - Frame formats

Frame Types	Link_Control (FT-0)	Acknowledge (ACK)	ACK_0 (bits 15-0 = 0)
			ACK_1 (bits 15-0 = 1)
			ACK_N (bits 15-0 = N)
		Link_Response	Busy F_BSY, P_BSY
			Reject F_RJT, P_RJT
		Link_Command	LCR
	Data (FT-1)	FC-4 Device_Data	FC-4 Device TYPEs IP, IPI-3, SCSI, SB, ...
		FC-4 Video_Data	FC-4 Video TYPEs Reserved
		Link_Data	Basic Link Service ABTS, BA_ACC, BA_RJT, NOP, RMC
			Extended Link Service ABTX, ACC, ADVC, ECHO, ESTC, ESTS FLOGI, LOGO, LS_RJT, PLOGI, RCS RES, RLS, RRQ, RSI, RSS, RTV, TEST
			FC-4 Link Service IP, IPI-3, SCSI, SB, ...

ACK and Link_Response frames are individual frames which indicate successful or unsuccessful frame delivery of a valid frame to the FC-2 level at the destination N_Port which also participate in end-to-end flow control. Successful delivery to the N_Port shall be indicated by ACK frames, while unsuccessful delivery shall be indicated by Link_Response frames. The R_RDY Primitive Signal is used for buffer-to-buffer flow control which is discussed in clause 26.

A set of one or more Data frames related by the same SEQ_ID transmitted unidirectionally from one N_Port to another N_Port is called a Sequence. See clause 24 for a discussion of Sequences and Exchanges.

Class 1 Data frames except a Class 1 connect-request shall be retransmitted, only if the Discard multiple Sequences with immediate retransmission Exchange error policy is in effect and the Sequence Recipient has requested Sequence retransmission on an ACK frame. Regardless of the error policy, a Class 1 connect-request or Class 2 Data frame shall be retransmitted, only in response to a corresponding Busy (F_BSY, P_BSY) frame. Except as above, Data frame recovery shall be by means of Sequence retransmission under the control of FC-4. See 29.6.2, 29.6.3, and 29.7 for a discussion of Sequence integrity, Sequence error detection, and Sequence recovery.

Each Data frame within a Sequence shall be transmitted within an E_D_TOV timeout period to avoid timeout errors at the destination N_Port.

Delimiters:
Table 48 indicates allowable delimiters for valid Data frames by class.

Table 48 - Data frame delimiters

Data frame	Delimiters
Class 1	SOF_{c1}, SOF_{i1}, SOF_{n1}, EOF_n
Class 2	SOF_{i2}, SOF_{n2}, EOF_n
Class 3	SOF_{i3}, SOF_{n3}, EOF_n, EOF_t

Format: FT_1 (See 17.4)

Addressing: The S_ID field designates the source N_Port identifier (Sequence Initiator) transmitting the Data frame. The D_ID field designates the destination N_Port identifier (Sequence Recipient) of the Data frame.
Data Field: The Data_Field for Data frames is a multiple of four bytes and variable in length. The Data Field may contain optional headers whose presence is indicated by the DF_CTL field in the Frame_Header (see clause 19).

In order to accommodate message content within the Data field that is not a multiple of four bytes, fill bytes shall be appended to the end of the Data Field. The number of fill bytes is specified by F_CTL bits 1-0 (see 18.5) and shall only be meaningful on the last frame of an instance of an Information Category. The fill byte value is not specified by FC-PH.

Payload size
Payload size is determined by taking the overall frame length between the **SOF** and **EOF**

minus the 24 byte Frame_Header,
minus any Optional Headers,
minus the fill bytes (0, 1, 2, or 3),
minus the CRC.

Responses:

R_RDY Primitive (**SOF_{c1}, SOF_{x2}, SOF_{x3}**)
ACK
— ACK_0
— ACK_1
— ACK_N
Link_Response
— F_RJT, P_RJT
— F_BSY, P_BSY

20.2.1 R_RDY response

In Class 1, a connect-request (**SOF_{c1}**) frame shall be responded to by transmitting the R_RDY Primitive Signal. The R_RDY Primitive Signal shall only be used for flow control and shall not indicate that the requested Dedicated Connection has been made. In Class 2 and Class 3, Data and Link_Control frames received shall be responded to by transmitting the R_RDY Primitive Signal. R_RDY transmission shall indicate that the interface buffer which received the frame is available for further frame reception.

20.2.2 Data frame resp nses

Link_Control response frames are ACK frames (0, 1, and N) and Link_Response frames (P_BSY, P_RJT, F_BSY, and F_RJT). All Link_Control response frames (ACK_0, ACK_1, ACK_N, P_BSY, F_BSY, P_RJT, and F_RJT) shall be transmitted in the same Class of Service as the frame to which it is responding. However, the ACK (ACK_1, ACK_N, or ACK_0) to a connect-request frame (SOFc1) is sent as a Class 1 frame and is not subject to buffer to buffer to flow control.

20.2.2.1 ACK frames

Successful Data frame delivery

Class 1
- ACK_0
- ACK_1
- ACK_N

Class 2
- ACK_0
- ACK_1
- ACK_N

Class 3
- No response

20.2.2.2 Link_Response frames

Unsuccessful Data frame delivery

Class 1
- F_BSY (Fabric Busy), or
- P_BSY (N_Port Busy), or
- F_RJT (Fabric Reject), or
- P_RJT (N_Port Reject).

Class 2
- F_BSY (Fabric Busy), or
- P_BSY (N_Port Busy), or
- F_RJT (Fabric Reject), or
- P_RJT (N_Port Reject).

Class 3
- No response

20.2.3 Transfer mechanisms

Several mechanisms are available for a Sequence which allow an FC-4 or Upper Level Protocol to convey information to a destination N_Port. Those mechanisms include:

- Information Category within R_CTL field
- Options within F_CTL field
- Device Header
- Payload

Information Category:

The Information Category is included in R_CTL to assist the receiver of a Data frame in directing the Data Field content to the appropriate buffer pool.

F_CTL field options:

Within the F_CTL field, Exchange and Sequence CTL bits are used to manage the initiative to transmit Data frames, manage Sequences, and manage Exchanges.

Device Header:

Provisions have been made for a Device_Header to precede the actual Payload contained in the Data Field of a Data frame. The size of the Device_Header is encoded in the DF_CTL field of the Frame_Header. This provides an additional means to specify unique protocol-dependent information to an upper level.

Payload

The normal method of locating a ULP header as part of the Payload provides a straightforward approach to achieve interoperability.

20.3 Link_Control

Link_Control frames shall be used by the N_Port link facility functions to control frame transfer and provide N_Port control in Class 1 and 2.

Link_Control frames are identified by the Routing bits 31-28 in the R_CTL field being set to 1 1 0 0. When a Link_Control frame is identified in R_CTL bits 31-28, the Information Category bits (27-24) contain the command code for each Link_Control frame as shown in table 49. The TYPE field for Link_Control frames other than F_BSY shall be reserved. The DF_CTL field shall be set to zeros since a Link_Control frame contains a Data Field of zero length.

Table 49 - Link Control codes		
Encoded Value Word 0, bits 27-24	Description	Abbr.
0000	Acknowledge_1	ACK_1
0001	Acknowledge_N Acknowledge_0	ACK_N ACK_0
0010	N_Port Reject	P_RJT
0011	Fabric Reject	F_RJT
0100	N_Port Busy	P_BSY
0101	Fabric Busy to Data frame	F_BSY
0110	Fabric Busy to Link_Control frame	F_BSY
0111	Link Credit Reset	LCR
Others	reserved	

Link_Control frames provide:

- indication of successful delivery,
- indication of unsuccessful delivery,
- flow control and buffer management feedback.
- low-level control commands to an N_Port.

An N_Port shall provide sufficient resources to receive Link_Control frames in response to Data frames transmitted. An N_Port shall not transmit P_BSY response frames in response to Link_Control frames.

NOTE - It is not necessary to save information in order to retransmit a Link_Control frame since F_BSY to a Link Control frame contains all information required to retransmit and P_BSY is not allowed for Link_Control frames.

LCR may always be retransmitted in response to an F_BSY. For ACK (0, 1, N) and RJT frames, see individual commands for any restrictions on frame retransmission in response to F_BSY. Link Control frames shall be transmitted within an E_D_TOV timeout period of the event which causes transmission of the Link Control frame.

Delimiters:
Table 50 indicates allowable delimiters for valid Link_Control frames by class.

Tabl 50 - Link_C ntr l frame delimiters	
ACK, BSY, RJT	Delimiters
Class 1	SOF_{n1}, EOF_n, EOF_t, EOF_{dt}
Class 2	SOF_{n2}, EOF_n, EOF_t
LCR	Delimiters
Class 2	SOF_{n2}, EOF_n

20.3.1 R_RDY response

In Class 2, all Link_Control frames shall be responded to by transmitting the R_RDY Primitive Signal when the interface buffer which received the frame is available for further frame reception.

20.3.2 Link_Continue function

The Link_Continue function in FC-PH provides a positive feedback mechanism to control the flow of Data frames on the link. A Data frame shall only be transmitted when the attached Port has indicated that a buffer is available for frame reception. The following list specifies flow control elements:

- R_RDY - buffer-to-buffer flow control for frames between F_Ports and N_Ports if a Fabric is present, or between N_Ports without a Fabric present. The R_RDY Primitive is transmitted on receipt of any Class 2 or 3 frame as well as a Class 1 connect-request frame.
- ACK_0 - successful or unsuccessful delivery of a Sequence (see 20.3.2.2) between N_Ports with or without a Fabric present. ACK_0 shall be applicable only to Class 1 and Class 2 Sequences.
- ACK_1 - end-to-end flow control for a single Data frame transfer between N_Ports with or without a Fabric present. The ACK_1 frame is transmitted on receipt of Class 1 or 2 Data frames.
- ACK_N - end-to-end flow control for one or more consecutive Data frame transfers betw en N_Ports with or without a Fabric present. The ACK_N frame is transmitted on receipt of Class 1 or 2 Data frames.

An N_Port should transmit R_RDY and Link_Control frames before Data frames in order to avoid Credit problems (both buffer-to-buffer and end-to-end). When both an R_RDY and a Link_Control frame (ACK, BSY, RJT) are being transmitted in response to a Data frame (Class 2 and Class 1 connect-request), the R_RDY shall be transmitted prior to the Link_Control frame.

ACK (0, 1, N) may be used for acknowledgment of Data frames between N_Ports for a given Sequence, but usage shall follow the allowable forms based on support defined in Login. Prior to N_Port Login, ACK_1 shall be used. Following N_Port Login, the decision to use ACK_0, ACK_1 or ACK_N is made based on the results of N_Port Login.

ACK precedence
When multiple ACK forms are supported by both Sequence Initiator N_Port Login parameters and the destination N_Port Sequence Recipient N_Port Login parameters, ACK_0 usage shall take precedence over ACK_N and ACK_N usage shall take precedence over ACK_1. ACK_1 shall be the default, if no other ACK form is supported by both ends. Mixing ACK forms within a given Sequence is not allowed (i.e., only one ACK form shall be used within a single Sequence). ACK precedence is summarized in figure 53.

		ACK support by Sequence Recipient at Login			
		ACK_1	ACK_N ACK_1	ACK_0 ACK_1	ACK_0 ACK_N ACK_1
ACK support by Sequence Initiator at Login	ACK_1	ACK_1	ACK_1	ACK_1	ACK_1
	ACK_1 ACK_N	ACK_1	ACK_N	ACK_1	ACK_N
	ACK_1 ACK_0	ACK_1	ACK_1	ACK_0	ACK_0
	ACK_1 ACK_N ACK_0	ACK_1	ACK_N	ACK_0	ACK_0

Figure 53 - ACK precedence

20.3.2.1 Receiver Ready (R_RDY)

The R_RDY Primitive shall indicate that a single frame (valid or invalid) was received and that the interface buffer which received the frame is available for further frame reception. An N_Port or F_Port may choose to support multiple buffers for frames using the R_RDY Primitive for flow control. The number of buffers is specified during Login as buffer-to-buffer Credit.

R_RDY shall be transmitted between an N_Port and an F_Port with a Fabric present, or between two N_Ports without a Fabric present for all Class 2 frames, Class 3 frames, and connect-request frames in Class 1. The R_RDY Primitive is not forwarded to any higher levels within an N_Port or passed beyond the entry or exit point of the Fabric.

See 16.3.2 for specification of the number of Idles before and after the R_RDY Primitive during transmission.

Responses: There is no response to an R_RDY Primitive.

20.3.2.2 Acknowledge (ACK)

The ACK frame shall indicate that one or more valid Data frames were received by the destination N_Port for the corresponding Sequence_Qualifier and SEQ_CNT of a valid Exchange as specified in the Sequence_Qualifier, and that the interface buffers which received the frames or frame are available for further frame reception. ACK frames shall be used in Class 1 and 2 and shall be transmitted in the same Class as the Data frame or frames which are being acknowledged.

NOTE - In Class 1, it is recommended that N_Ports transmit ACK_1 or ACK_N in the same order that the corresponding Data frames are received.

ACK_1

The ACK_1 form of ACK shall be supported by all N_Ports as the default. The SEQ_CNT of the ACK_1 shall match the single Data frame being acknowledged. The Parameter Field contains a valu of binary one in ACK_CNT (bits 15-0) to indicate that a single Data frame is being acknowledged. Th R_CTL field (Word 0, bits 27-24) shall be set to 0000.

ACK_0

ACK_0 is the designation used when the ACK_CNT (bits 15-0) of the Parameter Field of the ACK_0 frame contains a value of binary zero to indicate that all Data frames of a Sequence are being acknowledged. If the SEQ_CNT is allowed to wrap within a single Sequence, frame uniqueness is not being ensured. The SEQ_CNT of the ACK_0 shall match the SEQ_CNT of the last Data frame transmitted within the Sequence. The R_CTL field (Word 0, bits 27-24) shall be set to 0001.

The ACK_0 frame may be used for both Discard and Process Exchange Error Policies. For both policy types, ACK_0 support as indicated by Login also specifies that infinite buffering shall be used.

When multiple ACK forms are supported by both Sequence Initiator N_Port Login parameters and the destination N_Port Sequence Recipient N_Port Login Parameters, ACK_0 usage shall take precedence over ACK_N and ACK_N shall take precedence over ACK_1. ACK_1 shall be the default, if no other common ACK form is supported by both ends.

If ACK_0 is supported by both Sequence Initiator and Sequence Recipient, a single ACK_0 per Sequence shall be used to indicate successful Sequence delivery or to set Abort Sequence Condition bits. An additional ACK_0 shall be used within a Sequence to

a) perform X_ID interlock or
b) respond a connect-request (SOFc1).

ACK_0 shall not participate in end-to-end Credit management. Mixing ACK forms in a Sequence is not allowed.

NOTE - Although infinite buffers is indicated at the FC-PH level within an N_Port, individual FC-4s such as SCSI or IPI-3, for example, may agree on a maximum Sequence size that is specified at the upper level through Mode Select in SCSI and Attributes in IPI-3. In each protocol, a burst size is specified which indicates the largest single Sequence which shall be transferred. By further controlling the maximum number of concurrent Sequences, each N_Port may limit the amount of buffering that is actually required.

ACK_N

ACK_N is the designation used when th ACK_CNT (bits 15-0) of the Parameter Field contain a value of N (where N is 1 to 65535) in which the Recipient is acknowledging N consecutive Data frames of a Sequence with the SEQ_CNT of the ACK_N frame indicating the highest SEQ_CNT being acknowledged. For example, a value of N = 2 and a SEQ_CNT = 18 shall indicate that Data frames with SEQ_CNT = 17 and 18 are being acknowledged. The R_CTL field (Word 0, bits 27-24) shall be set to 0001. Support for the ACK_N frame is optional and is specified in the Service Parameters during Login (see 23.6).

NOTE - The Sequence Recipient sends ACK_1 or ACK_N at least once with Abort Sequence Condition bits set to a value other than 0 0 (see 24.3.10.2, 24.3.10.3, or 24.3.10.4).

All ACK forms

For all forms of ACK, when the History bit (bit 16) of the Parameter Field is set = 0, it shall indicate that the Sequence Recipient N_Port has transmitted all previous ACKs (i.e., lower SEQ_CNT), if any, for this Sequence. When the History bit (bit 16) of the Parameter Field is set = 1, it shall indicate that at least one previous ACK has not been transmitted (withheld, Data frame not processed, or Data frame not received) by the Sequence Recipient N_Port. Using this historical information allows an N_Port to reclaim end-to-end Credit for a missing ACK frame.

Being able to reclaim end-to-end Credit does not relieve the N_Port of accounting for all ACK frames of a Sequence in Class 2. ACK frames shall not be retransmitted in response to an F_BSY (Class 2). The F_BSY frame to an ACK shall be discarded.

Support for ACK_N and ACK_0 may not be symmetrical for a single N_Port as a Sequence Initiator and Sequence Recipient (see 23.6.8.3 and 23.6.8.4).

Note - Throughout FC-PH, ACK refers to one of the three forms (ACK_1, ACK_0, or ACK_N) and although there are two command codes in R_CTL, the Parameter Field History bit (bit 16) and ACK_CNT (bits 15-0) are used in a consistent manner.

The ACK frame provides end-to-end flow control for one or more Data frames between two N_Ports as defined in ACK_0, ACK_1 or ACK_N. See 26.4.3.2 for usage rules and annex O for examples. A specific Data frame shall be acknowledged once and only once. ACK reception does not imply Data delivery to a higher level.

ACK frames participate in a number of functions in addition to end-to-end flow control. The following list identifies many of those functions:

- X_ID assignment (see 24.4),
- X_ID reassignment (see 25.3.1),
- X_ID interlock (see 24.5.4),
- terminating a Sequence (see 24.3.8),
- establishing a Dedicated Connection (see 28.4.1),
- removing a Dedicated Connection (see 28.4.3),
- Abort Sequence condition (see 18.5 and 24.3.10),
- Stop Sequence condition (see 18.5 and 24.3.10),
- Abnormal Sequence termination (see 29.7.1), and
- Sequence retransmission requested (see 29.7.1.2).

Format: FT_0
Addressing: The D_ID field designates the source of the Data frame (Sequence Initiator) being replied to by the ACK, while the S_ID field designates the source of the ACK frame (Sequence Recipient).
F_CTL: The F_CTL field is returned with both Sequence and Exchange Context bits inverted in the ACK frame. Other bits may also be set according to table 39.
SEQ_ID: the SEQ_ID shall be equal to the SEQ_ID of the frame being replied to by ACK.
SEQ_CNT: The sequence count (SEQ_CNT) shall be equal to the sequence count of the highest Data frame being replied to by the ACK.
Parameter field: The Parameter Field is defined as follows:

- History Bit (bit 16)
  0 = all previous ACKs transmitted
  1 = at least one previous ACK not transmitted
- ACK_CNT (bits 15 - 0)
  0 = all Data frames (ACK_0)
  N = 1 to N Data frames (ACK_1, ACK_N)

Responses:

R_RDY Primitive (**SOFx2**)
Link_Response
- F_RJT, P_RJT
- F_BSY

See annex O for examples of use.

20.3.3 Link_Response

Link_Response frames shall be sent by either the destination N_Port or an F_Port in reply to frames for Class 1 and 2. Link_Response frames shall only be sent by an N_Port in reply to valid frames (see 17.8.1).

A Link_Response indicates that the frame identified by the Sequence_Qualifier and SEQ_CNT was not delivered to or processed by the destination N_Port. When a Link_Response frame (either Reject or Busy) is generated by an F_Port or a facility internal to the Fabric, the frame is routed to the destination N_Port of the Link_Response frame. Link_Response frames may be:

- Busy - indicates a Busy condition was encountered in the Fabric or destination N_Port.
- Reject - indicates that delivery of the frame is being denied.

20.3.3.1 Fabric Busy (F_BSY)

The F_BSY Link_Response shall indicate that the Fabric or the destination N_Port is temporarily occupied with other link activity and the Fabric is unable to deliver the frame. A reason code is identified in bits 31-28 of the TYPE field. In Class 1, F_BSY shall only be transmitted in response to the **SOFc1** frame and shall be ended with **EOFdt**. In Class 2, any Data frame or ACK frame may receive an F_BSY response. A Busy response shall not be used in Class 3.

There are two different Link_Control codes defined for F_BSY as shown in table 49. When word 0, bits 27-24 has a value of 0101, the F_BSY is in response to a Data frame. When word 0, bits 27-24 has a value of 0110, F_BSY is in response to a Link_Control frame.

An F_BSY frame shall not be transmitted in response to another Busy frame (either F_BSY or P_BSY); the Busy frame shall be discarded.

When an N_Port receives F_BSY in response to a Data frame transmission, the N_Port shall retransmit the busied frame if it has not reached its ability to retry. Therefore, an N_Port shall save sufficient information for Data frames with an **SOFc1**, or **SOFx2** delimiter for retransmission until an ACK or RJT is received or retry is exhausted.

If an N_Port has exhausted its ability to retry Data frame transmission in response to an F_BSY, it shall notify the FC-4 or an upper level. The N_Port may perform the Abort Sequence Protocol based on the Exchange Error Policy.

It is not necessary to save information in order to retransmit a Link_Control frame since F_BSY to a Link Control frame contains all information required to retransmit and P_BSY is not allowed for Link_Control frames. In Class 2, if an N_Port receives an F_BSY in response to an ACK frame, it shall discard the F_BSY frame.

The Fabric shall interchange the S_ID and D_ID fields, invert both the Exchange and Sequence Context bits, and select the correct Link_Control code value for the F_BSY depending on whether it is in response to a Data frame or Link_Control frame. The Parameter field is returned unchanged.

A reason code is indicated in the TYPE field (Word 2, bits 31-28). If the frame being busied is a Link Control frame, the Link_Control command code of the busied frame in R_CTL (Word 0, bits 27-24) is moved to the TYPE field (Word 2, bits 27-24) of the F_BSY frame before the F_BSY command code is inserted in R_CTL bits 27-24 of the F_BSY frame. The Data Field of a Data frame is discarded. The Fabric shall use **EOFdt** for Class 1 F_BSY frames (connect-request) and **EOFn** for Class 2 F_BSY frames.

Format: FT_0
Addressing: The D_ID field designates the source of the frame encountering the busy condition while the S_ID field designates the destination of the frame encountering the busy condition.
F_CTL: The F_CTL field is returned as received with both Sequenc and Exchange Context bits inverted in the F_BSY or P_BSY frame.
SEQ_ID: The SEQ_ID shall be equal to the SEQ_ID of the frame being busied.
SEQ_CNT: The SEQ_CNT shall be equal to the SEQ_CNT of the frame encountering the busy condition.
Parameter field: The Parameter field is returned unchanged for both Data frames and Link_Control frames.
TYPE field:
The F_BSY reason code format is defined in figure 54.

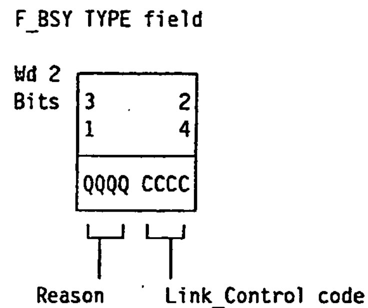


Figure 54 - F_BSY TYPE field

Table 51 - F_BSY reason codes

Encoded Value Wd 2, bits 31-28	Description
0001	Fabric busy
0011	N_Port busy
Others	Reserved

Busy reason codes are defined as follows:

Fabric busy

The Fabric is unable to deliver the frame to the destination N_Port due to conditions internal to the Fabric.

N_Port Busy

The destination N_Port is currently involved in a Class 1 Dedicated Connection and the Fabric is unable to deliver the frame.

Responses:

R_RDY Primitive (**SOFx2**)
Link_Response
— none

20.3.3.2 N_Port Busy (P_BSY)

The P_BSY Link_Response shall indicate that the responding N_Port is temporarily occupied with other link activity and is not able to accept the frame. A reason code shall be identified in the Parameter field of a P_BSY frame. In Class 1, P_BSY shall only be transmitted in response to the **SOF_{c1}** frame and shall be ended with **EOF_{dt}**. In Class 2, any Data frame may receive a P_BSY response. A Busy response shall not be used in Class 3.

A P_BSY frame shall not be transmitted in response to another Busy frame (either F_BSY or P_BSY); the Busy frame shall be discarded.

When an N_Port receives P_BSY in response to a frame transmission, the N_Port shall retransmit the busied frame if it has not reached its ability to retry. Therefore, a Port shall save sufficient information for Data frames with an **SOF_{c1}**, or **SOF_{i2}** delimiter for retransmission until an ACK or RJT is received or retry is exhausted.

If an N_Port has exhausted its ability to retry Data frame transmission in response to a P_BSY, it shall notify the FC-4 or an upper level. The N_Port may perform the Abort Sequence protocol based on the Exchange Error Policy.

N_Port Busy indicates that the Busy was issued by the destination N_Port. P_BSY shall not be issued in response to a Link_Control frame. An N_Port shall process a Link_Control frame for each unacknowledged Data frame transmitted.

Format: FT_0
Addressing: The D_ID field designates the source of the frame encountering the busy condition while the S_ID field designates the destination of the frame encountering the busy condition.
F_CTL: The F_CTL field is returned as received with both Sequence and Exchange Context bits inverted in the P_BSY frame. Other bits may also be set according to table 39.
SEQ_ID: The SEQ_ID shall be equal to the SEQ_ID of the frame being busied.
SEQ_CNT: The SEQ_CNT shall be equal to the SEQ_CNT of the frame encountering the busy condition.

Parameter field: The four bytes of this field indicate the action and reason code for the P_BSY response as defined in figure 55. Tables 52 and 53 specify Busy action and reason codes.

P_BSY parameter

	First Byte	Second Byte	Third Byte	Fourth Byte
Bits	31 … 24	23 … 16 (111 876)	15 … 8	7 … 0
	AAAA AAAA	CCCC CCCC	rrrr rrrr	VVVV VVVV
	Action	Reason		Vendor Unique

Figure 55 - P_BSY code format

Table 52 - P_BSY action codes

Encoded Value Wd 5, bits 31-24	Description
0000 0001 (1)	Sequence terminated - retry later
0000 0010 (2)	Sequence active - retry later
Others	Reserved

Action 1

Action 1 shall indicate that the Sequence Recipient has busied the Sequence (**EOF_t** or **EOF_{dt}**). For a Class 1 connect-request the busy frame is ended with **EOF_{dt}**. The Sequence Recipient shall only terminate the Sequence on a Busy in response to a connect-request (**SOF_{c1}**) or in response to an interlocked Data frame associated with X_ID assignment or reassignment (**SOF_{i2}**). The frame and Sequence are retryable at a later time.

Action 2

Action 2 shall indicate that the Sequence Recipient has busied a Class 2 frame and that the Sequence has not been terminated (**EOF_n**). The frame is retryable at a later time.

Table 53 - P_BSY reason codes	
Encoded Value Wd 5, bits 23-16	**Description**
0000 0001	Physical N_Port busy
0000 0011	N_Port Resource busy
1111 1111	Vendor Unique Busy (See Bits 7-0)
Others	Reserved

Busy reason codes are defined as follows:

Physical N_Port busy

P_BSY - the destination N_Port link facilities are currently busy and the N_Port is unable to accept the Payload of the frame.

N_Port Resource Busy

P_BSY - the destination N_Port is unable to process the Data frame at the present time.

Vendor Unique Busy

The Vendor Unique Busy bits (bits 7-0) shall be used by specific Vendors to specify additional reason codes.

Responses:

R_RDY Primitive **(SOFx2)**
Link_Response
— none

20.3.3.3 Reject (P_RJT, F_RJT)

The Reject Link_Response shall indicate that delivery of a frame is being denied. A four-byte reject action and reason code shall be contained in the Parameter field. Rejects are transmitted for a variety of conditions. For certain conditions retry is possible, whereas other conditions may require specific intervention. FC-PH does not define the specific intervention required.

In Class 1 and 2, if an error is detected by an N_Port or an F_Port in a Data frame, it shall transmit a Reject frame based on the reason codes specified in table 55. If an error is detected on a Link_Control frame, a Reject frame shall only be transmitted under specific conditions. A Fabric shall only r ject a Link_Control frame for the following reasons:

- Class not supported
- Invalid D_ID
- Invalid S_ID
- N_Port not available-temporary
- N_Port not available-permanent
- Login required (Fabric)

An N_Port shall only reject a Link_Control frame for the following reason (with Action code 2):

- Unexpected ACK

For other reasons, if an N_Port detects an error in a Link_Control frame for a valid Exchange, it shall initiate the Abort Sequence Protocol and not transmit a Reject frame. For an unidentified or invalid Exchange, if an error is detected in a Link_Control frame, the N_Port shall discard the frame and ignore the Link_Control frame error. If a Class 3 frame satisfies a rejectable condition, the frame shall be discarded. A Reject frame (F_RJT, P_RJT) shall not be transmitted in response to another Reject frame (either F_RJT or P_RJT); the received Reject frame in error shall be discarded.

Format: FT_0
Addressing: The D_ID field designates the source of the frame being rejected while the S_ID field designates the destination of the frame being rejected.
F_CTL: The F_CTL field is returned with both Sequence and Exchange Context bits inverted in the F_RJT or P_RJT frame. Other bits may also be set according to table 39.
SEQ_ID: The SEQ_ID shall be equal to the SEQ_ID of the frame being rejected.
SEQ_CNT: The SEQ_CNT shall be equal to the SEQ_CNT of the frame being rejected.
Parameter field: The four bytes of this field shall indicate the action code and reason for rejecting the request (see figure 56 and tables 54 and 55).

```
Reject parameter

             First     Second     Third     Fourth
             Byte       Byte      Byte       Byte
         ┌──────────┬──────────┬──────────┬──────────┐
Bits     │3        2│2      111│1         │          │
         │1        4│3      876│5        8│7        0│
         ├──────────┼──────────┼──────────┼──────────┤
         │AAAA AAAA │CCCC CCCC │rrrr rrrr │VVVV VVVV │
         └──────────┴──────────┴──────────┴──────────┘
           └──┬──┘    └──┬──┘               └──┬──┘
              │          │                     │
           Action     Reason             Vendor Unique
```

Figure 56 - Reject code f rmat

Tabl 54 - Reject action codes	
Encoded Value Wd 5, bits 31-24	Description
0000 0001 (1)	Retryable error
0000 0010 (2)	Non-retryable error
Other codes	Reserved

If a frame within a Sequence is rejected, the Sequence shall be abnormally terminated or aborted. If the RJT frame is ended by an **EOFt** or **EOFdt**, the Port transmitting the RJT frame has terminated the Sequence. A Port shall only terminate the Sequence using a Reject with an **EOFdt** in response to a connect-request (**SOFc1**). In Class 2 an N_Port shall only terminate the Sequence on a Reject in response to an interlocked Data frame associated with X_ID assignment or reassignment (**SOFi2**). If the RJT frame is ended by an **EOFn**, the N_Port receiving the RJT frame shall perform the Abort Sequence protocol to abort the Sequence. Rejects shall only be transmitted in response to valid frames.

Action 1

Action 1 indicates that if the condition indicated in the Reject Reason code is changed or corrected, the Sequence may be retryable.

- Class not supported
- Invalid D_ID
- Invalid S_ID
- N_Port not available-temporary
- N_Port not available-permanent
- Login required
- Excessive Sequences attempted
- Unable to establish Exchange

Applicability:

- by Fabric when D_ID = Fabric
- by Fabric when D_ID = N_Port
- by N_Port when D_ID = N_Port

Action 2

Action 2 indicates that the Sequence is non-retryable and further recovery such as Abort Exchange may be required. The following reason codes are non-retryable:

- Delimiter usage error
- Type not supported
- Invalid Link_Control
- Invalid R_CTL
- Invalid F_CTL
- Invalid OX_ID
- Invalid RX_ID
- Invalid SEQ_ID
- Invalid DF_CTL
- Invalid SEQ_CNT
- Invalid Parameter field
- Exchange error
- Protocol error
- Incorrect length
- Unexpected ACK
- Expiration_Security_Header not supported

Applicability:

- by Fabric when D_ID = Fabric
- by N_Port when D_ID = N_Port

The first error detected shall be the error reported; the order of checking is not specified.

Table 55 (Page 1 of 2) - Reject reason codes		
Encoded Value Wd 5, bits 23-16	Description	By
0000 0001	Invalid D_ID	B
0000 0010	Invalid S_ID	B
0000 0011	N_Port not available, temporary	F
0000 0100	N_Port not available, permanent	F
0000 0101	Class not supported	B
0000 0110	Delimiter usage error	B
0000 0111	TYPE not supported	B
0000 1000	Invalid Link_Control	P
0000 1001	Invalid R_CTL field	P
0000 1010	Invalid F_CTL field	P
0000 1011	Invalid OX_ID	P
0000 1100	Invalid RX_ID	P
0000 1101	Invalid SEQ_ID	P

Table 55 (Page 2 f 2) - Reject reason codes

Encoded Value Wd 5, bits 23-16	Description	By
0000 1110	Invalid DF_CTL	P
0000 1111	Invalid SEQ_CNT	P
0001 0000	Invalid Parameter field	P
0001 0001	Exchange Error	P
0001 0010	Protocol Error	P
0001 0011	Incorrect length	B
0001 0100	Unexpected ACK	P
0001 0101	Reserved	
0001 0110	Login Required	B
0001 0111	Excessive Sequences attempted	P
0001 1000	Unable to Establish Exchange	P
0001 1001	Expiration_Security_Header not supported	P
0001 1010	Fabric path not available	F
1111 1111	Vendor Unique Error (See Bits 7-0)	P
Others	Reserved	

NOTES
1 F = F_RJT (F_Port)
2 P = P_RJT (N_Port)
3 B = Both F_RJT and P_RJT

Invalid D_ID

F_RJT - the Fabric is unable to locate the destination N_Port address.

P_RJT - the N_Port which received this frame does not recognize the D_ID as its own Identifier.

Invalid S_ID

F_RJT - the S_ID does not match the N_Port Identifier assigned by the Fabric.

P_RJT - the destination N_Port does not recognize the S_ID as valid.

N_Port not available, temporary

F_RJT - The N_Port specified by the D_ID is a valid destination address but the N_Port is not functionally available. The N_Port is online and may be performing a Link Recovery Protocol, for example.

N_Port not available, permanent

F_RJT - The N_Port specified by the D_ID is a valid destination address but the N_Port is not functionally available. The N_Port is Offline or Powered Down.

Class not supported

F_RJT or P_RJT - The Class of Service specified by the **SOF** delimiter of the frame being rejected is not supported.

Delimiter usage error

F_RJT or P_RJT - The **SOF** or **EOF** is not appropriate for the current conditions. For example, a frame started by **SOFc1** is received while a Class 1 Dedicated Connection already exists with the same N_Port. See tables 48 and 50 for allowable delimiters by Class.

TYPE not supported

F_RJT or P_RJT - The TYPE field of the frame being rejected is not supported by the Port replying with the Reject frame.

Invalid Link_Control

P_RJT - The command specified in the Information Category bits within R_CTL field in the frame being rejected is invalid or not supported as a Link_Control frame.

Invalid R_CTL field

P_RJT - The R_CTL field is invalid or inconsistent with the other Frame Header fields or conditions present.

Invalid F_CTL field

P_RJT - The F_CTL field is invalid or inconsistent with the other Frame_Header fields or conditions present.

Invalid OX_ID

P_RJT - The OX_ID specified is invalid or inconsistent with the other Frame_Header fields or conditions present.

Invalid RX_ID

P_RJT - The RX_ID specified is invalid or inconsistent with the other Frame_Header fields or conditions present.

Invalid SEQ_ID

P_RJT - The SEQ_ID specified is invalid or inconsistent with the other Frame_Header fields or conditions present.

Invalid DF_CTL

P_RJT - The DF_CTL field is invalid.

Invalid SEQ_CNT

P_RJT - The SEQ_CNT specified is inconsistent with the other Frame_Header fields or conditions present. A SEQ_CNT reject is not used to indicate out of order or missing Data frames (see F_CTL Abort Sequence Condition bits 5-4).

Invalid Parameter field

P_RJT - The Parameter field is incorrectly specified or invalid.

Exchange Error

P_RJT - An error has been detected in the identified Exchange (OX_ID). This could indicate Data frame transmission without Sequence Initiative or other logical errors in handling an Exchange.

Protocol Error

P_RJT - This indicates that an error has been detected which violates the rules of FC-2 signaling protocol which are not specified by other error codes.

Incorrect length

F_RJT or P_RJT - The frame being rejected is an incorrect length for the conditions present.

Unexpected ACK

P_RJT - An ACK was received from an unexpected S_ID. The ACK received was not for an Open Sequence or Exchange, but was received from a Logged-in N_Port. (The **EOF** delimiter shall be **EOFn**.)

Login Required

F_RJT or P_RJT - An Exchange is being initiated before the interchange of Service Parameters (ie. Login) has been performed. F_RJT may be issued by the Fabric in order to notify an N_Port that a Login with the Fabric is required due to changes within the Fabric. F_RJT shall not be issued by the Fabric in order to convey Login status of a destination N_Port.

Excessive Sequences attempted

P_RJT - A new Sequence was initiated by an N_Port which exceeded the capability of the Sequence Recipient as specified in the Service Parameters during Login.

Unable to Establish Exchange

P_RJT - A new Exchange was initiated by an N_Port which exceeded the capability of the Responder facilities.

Expiration_Security_Header not supported

P_RJT - The N_Port does not support the optional Expiration_Security_Header.

Fabric path not available

F_RJT - The speed of the source and destination N_Ports does not match. Other Fabric characteristics related to multiple Fabric domains may also use this reason code.

Vendor Unique Reject

F_RJT or P_RJT - The Vendor Unique Reject bits (bits 7-0) shall be used by specific Vendors to specify additional reason codes.

Responses:

R_RDY Primitive (**SOFx2**)
Link_Response
— F_BSY

20.3.4 Link_Control commands

Link_Control commands are Link_Control frames which initiate a low-level action at the destination N_Port specified by the D_ID. These commands are limited in scope and are normally associated with functions such as reset. Link_Control commands do not require end-to-end Credit and do not participate in end-to-end flow control with regard to incrementing or decrementing Credit_CNT. Link_Control commands shall not be considered to be part of any existing Exchange or Sequence.

20.3.4.1 Link Credit Reset (LCR)

The Link Credit Reset frame shall indicate that the N_Port specified by the S_ID requests that the N_Port specified by the D_ID reset any buffers containing Data frames from the S_ID in order to allow the S_ID to reset its end-to-end Credit to its Login value, i.e., EE_Credit_CNT is set to zero. All Active Sequences with the S_ID as Sequence Initiator and the D_ID as Sequence Recipient are abnormally terminated for all classes by both N_Ports.

Exchange and Sequence recovery shall be performed at the discretion of the appropriate Upper Level Protocol by the S_ID N_Port. After transmitting the LCR frame, the N_Port which requested the Credit Reset shall wait the R_A_TOV timeout period before initiating Sequences with the destination N_Port. The Link Credit Reset frame shall not be transmitted as part of an existing Sequence or Exchange. All fields other than R_CTL, D_ID, and S_ID are reserved and ignored by the Receiver except for CRC calculation.

Link Credit Reset shall only be transmitted in Class 2. See 29.2.4.3 for a discussion of end-to-end Credit loss in Class 2 resulting from Sequence timeout. Any Class 1 and Class 3 Data frames in the destination N_Port buffers are also reset, although a Dedicated Connection, if present, is unaffected. Therefore, an N_Port should remove a Dedicated Connection prior to transmitting LCR in Class 2 to the same destination N_Port. LCR shall be transmitted with **SOFn2** and **EOFn**.

If an N_Port is out of Credit in Class 1 and times out all Sequences, it shall perform Connection Recovery (see 28.8). It shall not use LCR.

Format: FT_0
Addressing: The S_ID field designates the N_Port which is requesting a buffer reset by the destination N_Port or D_ID.
Responses:

R_RDY Primitive (**SOFn2**)
F_RJT, P_RJT
F_BSY

NOTE - F_RJT may be returned for any of the reasons allowed by the Fabric. P_RJT is only returned for "Invalid D_ID" or "Class not supported" in order to allow an N_Port to avoid special-casing LCR in Class 2. However, the N_Port transmitting LCR should be aware of possibility of F_RJT or P_RJT in order to avoid end-to-end Credit_CNT problems. In particular, the zero values of OX_ID, RX_ID, SEQ_ID, and SEQ_CNT should be noted for possible conflict with an existing Exchange.

20.4 Data frame summary

Table 56 summarizes Data frame responses by Class.

Table 56 - Data frame summary

DATA FRAME	FRAME RESPONSE (LINK_CONTROL)	
Data frame Sequence	**Expected (successful)**	**Alternate Link_Response (unsuccessful)**
Class 1		
Data frame	R_RDY Primitive to **SOFc1**, ACK_0, ACK_1, or ACK_N	F_BSY / PBSY or F_RJT / P_RJT
Class 2		
Data frame	R_RDY primitive plus ACK_0, ACK_1 or ACK_N	R_RDY primitive plus F_BSY / P_BSY or F_RJT / P_RJT
Class 3		
Data frame	R_RDY primitive	None

21 Link Services

21.1 Introduction

There are three types of Link Services:

- Basic,
- Extended, and
- FC-4.

Link Service frames and Sequences are composed of Link_Data frames and shall operate according to R_RDY Primitive, ACK, and Link_Response rules specified in clause 20. Since DF_CTL bits are not meaningful on Basic Link_Data frames, Basic Link Service frames shall not contain optional headers. The Expiration/Security Header is the only optional header allowed for Extended and FC-4 Link_Data frames.

21.1.1 Default Login values

Prior to Login or following Logout, a default set of Service Parameters apply:

- Concurrent Sequences = 1,
- Total Concurrent Sequences = 1,
- End-to-end Credit = 1,
- Buffer-to-buffer Credit = 1,
- Receive_Data_Field Size = 128,
- X_ID interlock required,
- no X_ID reassignment,
- ACK_1,
- Discard multiple Sequences Error Policy
- Relative Offset not used,
- Other optionally supported features shall not be used or required.

Following Login with the destination, an N_Port shall use the Service Parameters obtained through Login.

21.1.2 Sequence and Exchange management

Basic Link Service commands consist of only a single Basic Link_Data frame and are interspersed or are part of a Sequence for an Exchange performing a specific protocol other than Basic Link Service. In such cases, the Basic Link Service command does not constitute a separate Information Category in specifying the number of Information Categories in a Sequence as a Login parameter. Basic Link Service commands support low-level functions such as passing control bit information in a NOP, or aborting a Sequence using ABTS. Login shall not be required prior to using Basic Link Service commands.

The protocols supported by the Extended Link Services are performed within a single Exchange, intended exclusively for the purpose. Most of Extended Link Service protocols are performed as a two Sequence Exchange. Each of these two Sequence Exchanges consists of a request Sequence by the Originator (N_Port), transfer of Sequence Initiative, and a reply Sequence from the Responder (N_Port or F_Port) which terminates the Exchange by setting the Last_Sequence bit (Bit 20) in F_CTL.

The Sequence Initiator and Sequence Recipient shall follow the rules for Sequence management and Recovery_Qualifier reuse as specified in clause 24. The following rules regarding Sequence and Exchange management apply to Extended Link Services in addition to the rules specified in clause 24:

- Basic and Extended reply frames and Sequences shall be transmitted in the same Class as the request.
- if Login has not been completed successfully, each Port shall use the default Login values. Since Concurrent Sequences is limited to one, originating a second Exchange to perform an Extended Link Service, such as Read Exchange Status, is not possible. Therefore, Abort Sequence Protocol (ABTS) is the preferred recovery action.
- if Login has not been completed successfully, ESTC, ESTS, and ADVC protocols shall not be attempted or supported.
- if Login has been completed successfully, the Originator of the Exchange shall use the Discard multiple Sequences Error Policy for all Extended Link Service Exchanges.
- the Originator of an Extended Link Service Exchange shall detect an Exchange error following Sequence Initiative transfer if the Reply Sequence is not initiated and received within a timeout interval equal to twice the value of R_A_TOV.
- if the Exchang Originator of an Extended Link Service Exchange detects an Exchange error, it shall abort the Exchange (ABTX) and

retry the protocol of the aborted Exchange with a different Exchange.

— if the Sequence Initiator aborts a Sequence using ABTS (Abort Sequence Protocol) due to receiving an ACK with the Abort Sequence bits (5-4) set to 0 1, the Sequence Initiator shall retry the Sequence after the Basic Accept is received for the aborted Sequence one time only. If the retry fails, the Extended Link Service Exchange shall be aborted (ABTX).

— if the Sequence Initiator attempts to abort a Sequence using ABTS (Abort Sequence Protocol) and it detects a Sequence timeout (E_D_TOV) waiting for the ACK frame in response to the ABTS, it shall abort the Exchange (ABTX), if conditions permit, and retry the protocol of the aborted Exchange with a different Exchange (see 21.4.2).

21.2 Basic Link Service commands

All Basic Link Service commands (see table 57) shall be supported by an N_Port. Remove Connection (RMC) Link Service command is applicable to only Class 1. In a unidirectional Class 1 connection, any Basic Link Service request or reply may flow in either direction.

21.2.1 Routing control

R_CTL bits 31-28 (Word 0) are set = to 1000 to indicate a Basic Link_Data frame. The TYPE field (Word 2, bits 31-24) shall = 0000 0000 to indicate Basic Link Service. The command code of Basic Link Service commands shall be specified in the R_CTL bits 27-24. Table 57 specifies the Basic Link Service command codes.

Table 57 - Basic Link Service command codes

Encoded Value Word 0, bits 27-24	Description	Abbr.
0000	No Operation	NOP
0001	Abort Sequence	ABTS
0010	Remove Connection	RMC
0011	Reserved	
0100	Basic_Accept	BA_ACC
0101	Basic_Reject	BA_RJT
Others	reserved	

21.2.2 Ab rt Sequenc (ABTS)

The Abort Sequence Basic Link Service frame shall be used

a) by the Sequence Initiator to request that the Sequence Recipient abort one or more Sequences (see 21.2.2.1 and 29.7.1.1).
b) by the Sequence Initiator or Sequence Recipient to request that the ABTS Recipient abort the entire Exchange (see 21.2.2.2).

The decision to attempt to abort one or more Sequences may be determined by the Sequence Initiator (Sequence timeout) or the Sequence Recipient (ACK frame Abort Sequence Condition bits 5-4 or P_RJT frame). The Sequence Recipient may request that one or more Sequences in progress be aborted by setting the Abort Sequence Condition bits to a value of 0 1 on an ACK frame (see 18.5). The ABTS frame may be transmitted without regard to which N_Port holds, or may hold, the Sequence Initiative.

21.2.2.1 Aborting Sequences using ABTS

Using ABTS to abort one or more Sequences is specified in this section and 29.7.1.1. In this method,

— none, one or multiple Sequences are aborted;
— ABTS is transmitted by the Sequence Initiator of the last Sequence; and
— ABTS is transmitted as part of the Open Sequence.

The SEQ_ID of ABTS frame shall match the SEQ_ID of the last Sequence transmitted by the Sequence Initiator of the ABTS frame. Since ABTS is a continuation of the last transmitted Sequence, it shall be transmitted in the same Class of Service. Since Sequences shall not be streamed in more than one Class, the Class in which the ABTS is transmitted shall be the same Class in which an error, if any, occurred. The RX_ID and OX_ID specified in the ABTS Frame_Header shall be associated with the Exchange in which the Sequence Initiator has detected a potential error.

F_CTL bits, such as First_Sequence, shall be set to match previous Data frames within this Sequence since the ABTS frame is part of the Sequence. F_CTL bits for Sequence Initiative (bit 16) and End_Sequence (bit 19) shall be set to one in order to transfer S quence Initiative.

ABTS Initiator
Since ABTS is used for error recovery, the following relaxed behaviors are allowed. An ABTS Initiator may transmit ABTS, even if

- there is no end-to-end Credit available,
- it does not hold the Sequence Initiative,
- there is no Sequence Open,
- maximum number of Concurrent Sequences supported are in use, and
- it is a Connection Recipient in Unidirectional Class 1 Connection.

After transmitting the ABTS frame, an N_Port shall consider the status of the Exchange in which it was transmitted to be in an indeterminate condition and shall not deliver any Sequences or notification of Sequence delivery to an upper level until the Basic Accept is received, processed, and recovery, if any, is performed. Due to out of order delivery and special ACK transmission rules, an ACK to a Data frame within a Recovery_Qualifier range may mislead the Sequence Initiator of the ABTS prior to reception of the Basic Accept.

NOTE - The ABTS frame is allowed to be transmitted after a Sequence timeout. The Sequence Initiator of the ABTS frame should reset the E_D_TOV and R_A_TOV timers when the ABTS frame is transmitted, just as any other Data frame transmitted for a Sequence.

ABTS Recipient
When the ABTS request frame is received, the Sequence Recipient may abort no Sequences, one Sequence, or multiple Sequences based on the status of each Sequence within an Exchange and the Exchange Error Policy (see 29.6.1.1).: After receiving the ABTS frame, the Recipient shall determine a range of SEQ_CNT values found in error, if any, associated with the identified Exchange. Data frames for any deliverable Sequences may be processed after the ABTS frame is received based on the policy for the Exchange, but before the Basic Accept is transmitted.

Transmission of the Basic Accept to the ABTS frame is an atomic function in that any Data frames identified in the Recovery_Qualifier range (identified in Basic Accept Payload) shall be discarded after the Basic Accept is transmitted to the Sequence Initiator. The Basic Accept provides a synchronization point between the Sequ nce Initiator and Sequence Recipient. The ABTS Sequence Recipient is not required to timeout waiting for any missing frames before transmitting the Basic Accept. The ABTS Sequence Recipient shall set F_CTL bit 16 = 0 in the Basic Accept to indicate that it holds the Sequence Initiative for the Exchange or set it = 1 to indicate that the ABTS Sequence Initiator holds the Sequence Initiative.

The format of the Basic Accept Payload is shown in table 58. The SEQ_ID, if indicated as valid, shall be the last deliverable Sequence transmitted by the Sequence Initiator (of ABTS). If the SEQ_ID is indicated as invalid, then the Sequence Recipient has no information on the last deliverable Sequence. The low SEQ_CNT value shall be equal to the SEQ_CNT of the last Data frame of the last deliverable Sequence. The high SEQ_CNT value shall be equal to the SEQ_CNT of the ABTS frame.

In the Basic Accept Payload, if the low SEQ_CNT = high SEQ_CNT and the last valid SEQ_ID in the Basic Accept matches the last Sequence that was transmitted, then no Sequences have been aborted (all were deliverable), no Recovery_Qualifier is identified, and no recovery is required. If the low SEQ_CNT is not equal to the high SEQ_CNT or the last SEQ_ID is not the last Sequence transmitted, then at least one Sequence is in error.

Recovery Qualifier
If the ABTS frame was transmitted in Class 1, there shall be no Recovery_Qualifier. The Sequence Initiator of the ABTS frame may reuse SEQ_IDs at its discretion following the normal rules for SEQ_ID and SEQ_CNT and there is no timeout required.

If the ABTS frame was transmitted in Class 2 or 3 and a Sequence error has been indicated, a Recovery_Qualifier range shall be established for both N_Ports. If a Recovery_Qualifier exists, the Sequence Initiator of the ABTS frame shall discard ACK and Link_Response frames received which correspond to the Recovery_Qualifier between the low and high SEQ_CNT values. After transmission of the Basic Accept to the ABTS frame the Sequence Recipient of the ABTS frame shall discard Data frames received which correspond to the Recovery_Qualifier between the low and high SEQ_CNT values if a Recovery_Qualifier exists. While the Recovery_Qualifier exists, the Sequence Initiator shall not transmit Data frames

for the Recovery_Qualifier within the specified low and high SEQ_CNT values.

If a Recovery_Qualifier has been established, based on the Basic Accept Payload, the Sequence Initiator of the ABTS shall issue a Reinstate Recovery Qualifier (RRQ) Extended Link Service request Sequence after waiting an R_A_TOV timeout period after reception of the Basic Accept (see 21.4.14).

After the Basic Accept has been transmitted and the Sequence status has been posted in the Exchange Status Block as Aborted, if the Sequence Recipient receives any Data frames for the Aborted Sequence or Aborted Sequences (based on Recovery_Qualifier range), the frames shall be discarded. See 29.7.1 and 29.2.4 for more discussion on abnormal termination of Sequences and Sequence timeout. See 29.7.1.1 for examples of the ABTS protocol which include several special cases such as the start of an Exchange and Class 3. Additional information regarding Sequence recovery and the effects of ABTS based on different Exchange Error Policies is also discussed.

Following reception of the Basic Accept to the Abort Sequence frame, the Sequence Initiator may perform Sequence recovery under guidance from the appropriate FC-4.

The addressing is fully specified by the D_ID and S_ID fields.

Protocol:

Abort Sequence request frame
Basic Accept reply frame

Format: FT_1

Addressing: The D_ID field designates the destination N_Port (Sequence Recipient) while the S_ID field designates the source N_Port (Sequence Initiator) which is requesting that a Sequence or Sequences be aborted.

X_ID: Both the RX_ID and OX_ID shall correspond to the current values as determined by the Sequence Initiator of the ABTS frame.

SEQ_ID, SEQ_CNT: The SEQ_ID shall be the same as the last Sequence transmitted for this Exchange by the N_Port transmitting ABTS, even if the last Data frame has been transmitted. The SEQ_CNT shall be set to a value one greater than the previous Data frame transmitted, indicating the highest SEQ_CNT transmitted for this SEQ_ID and the highest SEQ_CNT for this range of SEQ_CNTs over multiple Sequences.

Payload: The Abort Sequence Basic Link Service command has no Payload.

Reply Sequence:

Basic Reject (BA_RJT)
signifies rejection of the ABTS command, however, the Sequence may have been aborted without Sequence information (see 21.2.4).

Basic Accept (BA_ACC)
signifies that the destination N_Port has aborted and discarded no Sequences, one Sequence, or multiple Sequences.

- BA_ACC Frame Header (see 21.2.3)
- Basic Accept Payload

The format of the Basic Accept Payload is shown in table 58.

The SEQ_ID, if indicated as valid, shall be the last deliverable Sequence transmitted by the Sequence Initiator. If the SEQ_ID is indicated as invalid, then the Sequence Recipient has no information on the last deliverable Sequence.

The high SEQ_CNT shall be equal to the SEQ_CNT of the ABTS frame. The low SEQ_CNT value shall be one of the following:

- same as SEQ_CNT of the ABTS frame,
- equal to the SEQ_CNT of the last Data frame of the last deliverable Sequence, or
- equal to hex '00 00'.

Payload is specified for each of the permitted cases.

1. To indicate that the current Sequence in which ABTS has been received is the last deliverable Sequence, and no Sequences are aborted at its end, the Sequence Recipient shall set, in the BA_ACC payload,

- SEQ_ID Validity = valid (hex '80')
- SEQ_ID = the SEQ_ID of the Sequence in which the ABTS has been received from the Sequence Initiator, and
- Low SEQ_CNT = High SEQ_CNT = SEQ_CNT of the ABTS frame

2. To indicate that it has the information on the last deliverable Sequence but one or more Sequences are aborted at its end, the Sequence Recipient shall set, in the BA_ACC payload,

- SEQ_ID Validity = valid (h x '80')
- SEQ_ID = the SEQ_ID of the last deliverable Sequence received from the

Sequence Initiator but is not equal to the SEQ_ID of the Sequence in which ABTS frame has been received,
- Low SEQ_CNT = the SEQ_CNT of the last Data frame of the last deliverable Sequence
- High SEQ_CNT = the SEQ_CNT of the ABTS frame

3. To indicate that it has no information on the last deliverable Sequence, and One or more Sequences are aborted at its end, the Sequence Recipient shall set, in the BA_ACC payload, independent of Continuously Increasing SEQ_CNT use,
- SEQ_ID Validity = invalid (hex '00')
- SEQ_ID = invalid in this case
- Low SEQ_CNT = hex '00 00'
- High SEQ_CNT = the SEQ_CNT of the ABTS frame.

21.2.2.2 Aborting an Exchange using ABTS

Using ABTS to abort an Exchange is specified in this section. In this method,

- an entire Exchange is aborted;
- ABTS is transmitted by the Sequence Initiator or the Sequence Recipient of the last Sequence; and
- ABTS is transmitted as part of the Open Sequence or in a new Sequence.

ABTS sent by the last Sequence Initiator in an Open Sequence
If the last Sequence is Open and the Sequence Initiator of the last Sequence transmits the ABTS frame, the SEQ_ID of this ABTS frame shall match the SEQ_ID of the last Sequence transmitted by the last Sequence Initiator. The SEQ_CNT of the ABTS frame shall be one greater than the SEQ_CNT of the last Data frame transmitted for this last Sequence.

ABTS sent by the last Sequence Initiator in a new Sequence
If the last Sequence has been completed and is therefore not Open, and the Sequence Initiator of the last Sequence transmits the ABTS frame, the ABTS shall be transmitted in a new Sequence with a valid SEQ_ID not in use at that time.

ABTS s nt in an Open or n w Sequence
Since ABTS is a continuation of the last transmitted Sequence, it shall be transmitted in the same Class of Service. Since Sequences shall not be streamed in more than one Class, the Class in which the ABTS is transmitted shall be the same Class in which an error, if any, occurred. The RX_ID and OX_ID specified in the ABTS Frame_Header shall be associated with the Exchange in which the Sequence Initiator has detected a potential error.

F_CTL bits for Sequence Initiative (bit 16) and End_Sequence (bit 19) shall be set to one in order to transfer Sequence Initiative. If the ABTS frame is part of the last Sequence, F_CTL bits, such as First_Sequence, shall be set to match previous Data frames within this Sequence. If the ABTS is transmitted in a new Sequence, F_CTL bits shall be set to match the new Sequence.

ABTS by the last Sequence Recipient
If the last Sequence Recipient chooses to transmit an ABTS frame, it shall transmit ABTS in a new Sequence with a SEQ_ID available for use from its N_Port as the Sequence Initiator. The new Sequence shall follow applicable rules for the Sequence. The Class in which the ABTS is transmitted shall be the same Class in which an error, if any, occurred. The RX_ID and OX_ID specified for the new Sequence shall be associated with the Exchange in which the Sequence Recipient has detected a potential error.

If the Sequence Initiator has not transferred the Sequence Initiative or has transferred the Sequence Initiative but has not received the confirmation, but receives the ABTS frame then the Sequence Initiator shall abort the Exchange. Aborting the Exchange may be performed by one of the following two methods specified:

On receiving the ABTS, the ABTS Recipient may set

a) the Last_Sequence bit to one in BA_ACC or
b) the Last_Sequence bit to zero in BA_ACC followed by ABTX command.

NOTE - If the Sequence Initiator has transferred the Sequence Initiative, received the confirmation but receives ABTS, then it is treated as the ABTS sent by the new Sequence Initiator and the corresponding rules are followed.

Protocol:

Abort Sequence request frame
Basic Accept reply frame

Format: FT_1

Addressing: The D_ID field designates the destination N_Port (ABTS Recipient) while the S_ID field designates the source N_Port (ABTS Initiator) which is requesting that an Exchange be aborted.

X_ID: Both the RX_ID and OX_ID shall correspond to the current values as determined by the Sequence Initiator of the ABTS frame.

SEQ_ID, SEQ_CNT:

1. If the Sequence Initiator is the ABTS initiator and a Sequence is Open, the SEQ_ID shall be the same as the last Sequence transmitted for this Exchange by the N_Port transmitting ABTS, even if the last Data frame has been transmitted. The SEQ_CNT shall be set to a value one greater than the previous Data frame transmitted, indicating the highest SEQ_CNT transmitted for this SEQ_ID and the highest SEQ_CNT for this range of SEQ_CNTs over multiple Sequences.
2. If the Sequence Initiator is the ABTS Initiator and no Sequence is Open, the SEQ_ID shall be a new valid value unused at that time and the SEQ_CNT shall be either continuously increasing from the latest Data frame transmitted in the last Sequence or binary zero.
3. If the Sequence Recipient is the ABTS Initiator, the SEQ_ID shall be a new valid value unused at that time by that N_Port as a Sequence Initiator and the SEQ_CNT shall be either continuously increasing from the latest Data frame transmitted in the last Sequence or binary zero.

Payload: The Abort Sequence Basic Link Service command has no Payload.

Reply Sequence:

Basic Reject (BA_RJT)
signifies rejection of the ABTS command, however, the Sequence may have been aborted without Sequence information (see 21.2.4).

Basic Accept (BA_ACC)
signifies that the destination N_Port has aborted and discarded no Sequences, one Sequence, multiple Sequences, or the entire Exchange.

- BA_ACC Frame Header (see 21.2.3)
- BA_ACC Payload
The format of the Basic Accept Payload is shown in table 58.
The SEQ_ID, if indicated as valid, shall be the last deliverable Sequence received from the Sequence Initiator. If the SEQ_ID is indicated as invalid, then the Sequence Recipient has no information on the last deliverable Sequence.

To abort an Exchange, the Last_Sequence bit shall be set to 1 and Low SEQ_CNT shall be hex '00 00' and High SEQ_CNT hex 'FF FF'. Payload for are specified for permitted cases:

1. To indicate that it has the information on the last deliverable Sequence, and nothing is aborted at its end, the ABTS Recipient shall set, in the BA_ACC payload,
- SEQ_ID Validity = valid (hex '80')
- SEQ_ID = the SEQ_ID of the last deliverable Sequence received from the ABTS Initiator, and
- Low SEQ_CNT = High SEQ_CNT = SEQ_CNT of ABTS frame

2. To indicate that it has no information on the last deliverable Sequence, and it is aborting the entire Exchange, the ABTS Recipient shall set the Last_Sequence F_CTL bit to 1 and shall set, in the BA_ACC payload,
- SEQ_ID Validity = invalid (hex '00'),
- SEQ_ID = invalid in this case
- Low SEQ_CNT = hex '00 00' and
- High SEQ_CNT = hex 'FF FF'

3. To indicate that it has information on the last deliverable Sequence, but it is aborting the entire Exchange due to uncertainty such as Sequence Initiative ownership, or lack of its capability to resolve the conflict, the ABTS Recipient shall set the Last_Sequence F_CTL bit to 1 and shall set, in the BA_ACC payload,
- SEQ_ID Validity = valid (hex '80')
- SEQ_ID = the SEQ_ID of the last deliverable Sequence received from the ABTS Initiator,
- Low SEQ_CNT = hex '00 00' and
- High SEQ_CNT = hex 'FF FF'.

Table 58 - ABTS Basic Accept Payload	
Item	Size -Bytes
SEQ_ID Validity (hex '80' = valid, hex '00' = invalid)	1
SEQ_ID of last Sequence deliverable to ULP (if valid indicated)	1
Reserved	2
OX_ID	2
RX_ID	2
Low SEQ_CNT	2
High SEQ_CNT	2

21.2.3 Basic Accept (BA_ACC)

The Basic Accept (BA_ACC) is a single frame Link Service reply Sequence that notifies the transmitter of a Basic Link Service request frame that the request has been completed. The Basic Accept Link Service reply Sequence shall transfer the Sequence Initiative by setting the Sequence Initiative bit (Bit 16) to 1 in F_CTL on the last Data frame of the reply Sequence if the Sequence Initiative for the Exchange is held by the transmitter of the ABTS frame. The Sequence Initiative (Bit 16) shall be set to 0 to indicate that the transmitter of the Basic Accept holds the Sequence Initiative for the Exchange. The OX_ID and RX_ID shall be set to match the Exchange in which the ABTS frame was transmitted. The SEQ_ID shall be assigned following the normal rules for SEQ_ID assignment.

Protocol:

Basic Accept (BA_ACC) is the reply Sequence to Abort Sequence Basic Link Service command.

Format: FT_1

Addressing: The D_ID field designates the source of the Link Service frame being accepted while the S_ID field designates the destination of the request Data frame Sequence being accepted.

Payload: The Payload content is defined within individual Basic Link Service command (ABTS).

Reply Sequence:

none

21.2.4 Basic Reject (BA_RJT)

The Basic Reject (BA_RJT) is a single frame Link Service reply Sequence that notifies the transmitter of a Basic Link Service request frame that the request has been rejected. A four-byte reason code is contained in the Payload. Basic Reject may be transmitted for a variety of conditions which may be unique to a specific Basic Link Service request. The OX_ID and RX_ID shall be set to match the Exchange in which the Basic request frame was transmitted. The SEQ_ID shall be assigned following the normal rules for SEQ_ID assignment.

Protocol:

BA_RJT may be a reply Sequence to ABTS.

Format: FT_1

Addressing: The D_ID field designates the source of the Basic Link Service request being rejected while the S_ID field designates the destination of the request Data frame Sequence being rejected.

Payload: The first word of the Payload shall contain four bytes to indicate the reason for rejecting the request (see figure 57 and tables 59 and 60).

```
Basic Application Reject data definition

          First      Second     Third      Fourth
          Byte       Byte       Byte       Byte
        +----------+----------+----------+----------+
Bits    |3        2|2      111|1         |          |
        |1        4|3      876|5        8|7        0|
        +----------+----------+----------+----------+
        |rrrr rrrr|CCCC CCCC|EEEE EEEE|VVVV VVVV|
        +----------+----------+----------+----------+
        Reserved    Reason     Reason     Vendor
                    Code       Explanation Unique
```

Figure 57 - BA_RJT format

Table 59 - BA_RJT reason codes	
Encoded Value (Bits 23-16)	Description
0000 0001	Invalid Command code
0000 0011	Logical error
0000 0101	Logical busy
0000 0111	Protocol error
0000 1001	Unable to perform command request
Others	Reserved
1111 1111	Vendor Unique Error (See Bits 7-0)

The first error condition detected shall be the error reported.

• Bits 23-16 Description

Invalid Command code

The Command code in the Sequence being rejected is invalid.

L gical rror

The r quest identified by the Command code is invalid or logically inconsistent for the conditions present.

Logical busy

The Basic Link Service is logically busy and unable to process the request at this time.

Protocol Error

This indicates that an error has been detected which violates the rules of FC-2 protocol which are not specified by other error codes.

Unable to perform command request

The Recipient of a Link Service command is unable to perform the request at this time.

Vendor Unique Error

The Vendor Unique Error bits may be used by Vendors to specify additional reason codes.

• Bits 15-8 Reason explanation

Table 60 shows expanded explanations for Basic Link Service commands.

Reply Link Service Sequence:

none

Table 60 - BA_RJT reason code explanation		
Encoded Value (Bits 15-8)	Description	Applicable commands
0000 0000	No additional explanation	ABTS
0000 0011	Invalid OX_ID-RX_ID combination	ABTS
0000 0101	Sequence Aborted, no Sequence information provided	ABTS
Others	Reserved	

21.2.5 No Operation (NOP)

The No Operation Basic Link Service frame shall be used with delimiters appropriate to the Class in which it is being used. The Data_Field of a NOP frame shall be of zero size. However, the F_CTL field and the **SOF** and **EOF** delimiters shall be examined and the appropriate action shall be taken by both the N_Port and Fabric, if present. A NOP frame may be used to initiate a Class 1 Connection using **SOFc1**, initiate Sequences, terminate Sequences, or terminate Class 1 Connections in place of a normal Data frame when there is no Data to send. When used to remove Class 1 Connections, NOP shall use the normal remove connection procedure by using the E_C bit (see 28.4.3).

The OX_ID and RX_ID shall be set to match the Exchange in which the NOP is being transmitted. The SEQ_ID shall be assigned following the normal rules for SEQ_ID assignment.
Protocol:

No Operation request
No reply frame

Format: FT_1
Addressing: The D_ID field designates the destination of the frame while the S_ID field designates the source of the frame.
Payload: The NOP Basic Link Service command has no Payload.
Reply Sequence:

none

21.2.6 Remove Connection (RMC)

The Remove Connection Basic Link Service frame shall be used to request immediate removal of a Class 1 Connection. An ACK frame ended by **EOFdt** shall be transmitted in response. This protocol overrides the normal negotiated remove connection procedure using the E_C bit in F_CTL. This method should not be used as the normal method to remove Class 1 Connections since all Open Sequences shall be abnormally terminated. See 28.4.3 for specific rules on removing connections and RMC.

The OX_ID and RX_ID shall be set to match th Exchange in which the RMC is being transmitted. The SEQ_ID shall be assigned following the normal rules for SEQ_ID assignment.

Protocol:

Remove Connection request
No reply frame

Format: FT_1
Addressing: The D_ID field designates the N_Port being requested to terminate the Class 1 Connection while the S_ID field designates the N_Port requesting removal.
Payload: The RMC Basic Link Service command has no Payload.
Reply Sequence:

none

21.3 Extended Link Services

An Extended Link Service request solicits a destination Port (F_Port or N_Port) to perform a function or service. The Information Category for a request is specified as Unsolicited Control. An Extended Link Service reply may be transmitted in answer to an Extended Link Service request. The Information Category for a reply is specified as Solicited Control. Each request or reply is composed of a single Sequence with the LS_Command code being specified in the first word of the Payload of the first frame of the Sequence.

Each Sequence may be composed of one or more frames. Normal rules for Exchange and Sequence management apply to Extended Link Service frames, Sequences, and Exchanges. The Accept to the following requests shall terminate the Exchange by setting the Last Sequence bit (Bit 20) to one on the last frame of the reply Sequence. The following Extended Link Service requests and the corresponding replies shall be performed within a single Exchange:

- Abort Exchange
- Echo
- Login
- Logout
- Read Connection Status
- Read Exchange Status Block
- Read Link Error Status Block
- Read Sequence Status Block
- Read Timeout Value
- Reinstate Recovery Qualifier
- Request Sequence Initiative
- Test

Establish Streaming, Estimate Credit, and Advise Credit may be in a single Exchange or multiple Exchanges.

21.3.1 Routing control

R_CTL bits 31-28 (Word 0) are set = to 0010 to indicate an Extended Link_Data frame. The TYPE field for Extended Link Service frames shall = 0000 0001. The Information Category bits shall be set to unsolicited control for request Sequences and solicited control for reply Sequences.

21.3.2 Extended Link Service command codes

When R_CTL bits 31-28 and TYPE indicate Extended Link Service, the first word of the Payload (LS_Command code) of the request or reply Sequence shall be encoded as shown in table 61 with bits 23-0 reserved. Subsequent frames, if any, for a request or reply Sequence shall only contain additional Payload in the Payload field (i.e. the LS_Command code is not repeated in each frame).

Table 61 - LS_Command codes

Encoded Value (Bits 31-24)	Description	Abbr.
0000 0001	Link Service Reject	LS_RJT
0000 0010	Accept	ACC
0000 0011	N_Port Login	PLOGI
0000 0100	F_Port Login	FLOGI
0000 0101	Logout	LOGO
0000 0110	Abort Exchange	ABTX
0000 0111	Read Connection Status	RCS
0000 1000	Read Exchange Status Block	RES
0000 1001	Read Sequence Status Block	RSS
0000 1010	Request Sequence Initiative	RSI
0000 1011	Establish Streaming	ESTS
0000 1100	Estimate Credit	ESTC
0000 1101	Advise Credit	ADVC
0000 1110	Read Timeout Value	RTV
0000 1111	Read Link Status	RLS
0001 0000	Echo	ECHO
0001 0001	Test	TEST
0001 0010	Reinstate Recovery Qualifier	RRQ
Others	Reserved	

21.4 Extended Link Service requests

A Sequence Initiator shall transmit a Link Service Sequence in order to solicit the destination N_Port to perform a link-level function or service. If an Extended Link Service request Sequence is transmitted without the transfer of Sequence Initiative, the Recipient shall abort the Exchange and not perform the request. An Extended Link Service Protocol is composed of an Extended Link Service request Sequence and an Extended Link Service reply Sequence. The last Data frame of an Extended Link S rvice request Sequence shall transfer the Sequence Initiative to the Recipient in order to allow the reply to be transmitt d (see clause 24). Th fol-

lowing Extended Link Service Protocols are defined:

- Abort Exchange (ABTX)
- Advise Credit (ADVC)
- Echo (ECHO)
- Establish Streaming (ESTS)
- Login (FLOGI/PLOGI)
- Logout (LOGO)
- Read Connection Status (RCS)
- Read Exchange Status Block (RES)
- Read Link Error Status Block (RLS)
- Read Sequence Status Block (RSS)
- Read Timeout Value (RTV)
- Reinstate Recovery Qualifier (RRQ)
- Request Sequence Initiative (RSI)

The following Extended Link Service requests have no reply Sequence:

- Estimate Credit
- Test

The following Extended Link Service requests and their replies shall be supported by an N_Port and others are optional.

FLOGI
PLOGI
LOGO
RRQ (Class 2 and 3 only)

An N_Port is not required to generate and send the PLOGI Extended Link Service request. However, if an N_Port receives PLOGI Extended Link Service request, the N_Port shall respond with the specified Link Service reply (ACC), or if PLOGI frame has invalid information, shall respond with LS_RJT. LS_RJT shall not be issued for the reason "PLOGI command not supported".

NOTE - If an N_Port which does not generate PLOGI, is in a point-to-point topology, and has an N_Port_Name greater than the other N_Port's, the other N_Port may timeout, waiting to receive PLOGI.

All Extended Link Service requests, excluding ESTS, ESTC, and ADVC, and the corresponding replies shall be performed within a single Exchange (see 23.8 for the procedure using ESTS, ESTC, and ADVC). The Advise Credit request may also be performed in a separate Exchange.

The Originator of the Link Service request shall assign an OX_ID when the request Sequence and Exchange are originated. If the only Exchange which the Responder has Open with the Originator's N_Port is the one for the Link Service being performed, a value of hex 'FFFF' may be used. The Responder shall assign an RX_ID when the request Sequence is received. A value of hex 'FFFF' may be used if only one Exchange is Open with the Originator.

Normal Sequence and Exchange management rules shall apply. See clause 24 for more information regarding use of F_CTL bits for Exchange and Sequence control as well as OX_ID, RX_ID, SEQ_ID, and SEQ_CNT.

Payload
Link Service requests and Link Service reply frames use an FT_1 frame type. The Payload shall contain a multiple of four bytes of data based on the individual Link Service request or reply. If required, more than one frame may be used to form a request or reply Sequence.

21.4.1 Link Service request collisions

There are two Extended Link Service request Sequences in which a collision is possible with the other N_Port involved in the same target Exchange. Those two requests are Abort Exchange (ABTX) and Request Sequence Initiative (RSI). For example, N_Port (A) may transmit an ABTX request to N_Port (B) at the same time that N_Port (B) transmits an ABTX request to N_Port (A) for the same target Exchange.

If such an instance occurs, the Originator N_Port of the target Exchange shall reject the ABTX or RSI request Sequence with an LS_RJT with a reason code of Command already in progress. The Responder N_Port of the target Exchange shall honor and process the ABTX or RSI request Sequence normally.

21.4.2 Abort Exchange (ABTX)

The Abort Exchange Link Service request shall be used to request abnormal termination of an Open Exchange, in progress. A request to abort an Exchange shall only be accepted if the request is made by the Originator N_Port or the Responder N_Port of the target Exchange.

A separate Exchange shall be used to abort the existing Exchange. The Payload shall contain the OX_ID and RX_ID for the Exchange being

aborted, as well as the S_ID of the N_Port which originated the Exchange.

Transmission of an ABTX frame is allowed while the identified Exchange is Open (see 24.3.14). Both the Originator and Responder shall ensure that the OX_ID and RX_ID pair being terminated are currently associated with the OX_ID and RX_ID pair specified in the ABTX request.

If an RX_ID other than hex 'FFFF' has not been received by the Sequence Initiator or the First Sequence of the Exchange has not completed successfully, the Abort Sequence Protocol shall be performed prior to attempting to Abort the Exchange using ABTX (see 29.7.1.1). This ensures that the Exchange Status Block exists and that both the OX_ID and RX_ID are assigned and known to both the Originator and Responder.

In Class 1, both the OX_ID and RX_ID are available by the respective N_Ports for reuse after the Exchange has been successfully aborted (Accept transmitted and received). If all frames are not accounted for in Class 2 or 3, a Recovery_Qualifier shall be established with a low SEQ_CNT of binary zero and a high SEQ_CNT of hex 'FFFF'.

The N_Port which issues the ABTX request shall set the Recovery_Qualifier status in the Payload of the ABTX request to indicate whether it believes a Recovery_Qualifier is required or not required. If the ABTX indicates that a Recovery_Qualifier is not required, the Recipient of the ABTX may agree by transmitting an Accept reply Sequence or it may disagree by transmitting an LS_RJT with the reason code of Recovery_Qualifier required.

If the ABTX requires a Recovery_Qualifier in its Payload and it is Accepted by the destination N_Port, then the N_Port which requested the ABTX shall transmit a Reinstate Recovery Qualifier (RRQ) Extended Link Service request following an R_A_TOV timeout in order to free those Exchange resources.

Any Active or Open Sequences associated with the Exchange are abnormally terminated by each N_Port. The ACC reply confirms that all Sequences and the Exchange have been abnormally terminated. It also confirms the presence or absence of a Recovery_Qualifier. If ABTX is used, then the use of ABTS to precede or follow ABTX is optional.

The addressing is fully specified by the D_ID and S_ID fields.

Protocol:

Abort Exchange request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The D_ID field designates the destination N_Port of the Exchange being aborted while the S_ID field designates the source N_Port which is requesting that the Exchange be aborted.

X_ID: A separate and distinct Exchange shall be required other than the Exchange being aborted in order to properly track status.

SEQ_ID, SEQ_CNT: The SEQ_ID and the SEQ_CNT shall be appropriate for an Active Sequence.

Payload: The format of the Payload is shown in table 62.

Table 62 - ABTX Payload

Item	Size -Bytes
hex '06000000'	4
Recovery_Qualifier status hex '00' no Recovery Qualifier hex '80' Recovery_Qualifier required	1
Originator S_ID	3
OX_ID	2
RX_ID	2
Association Header (optionally required)	32

If the Sequence Recipient has indicated that X_ID reassignment is required during Login, the Sequence Initiator shall include the Association Header in the Payload of the ABTX request associated with the Exchange being aborted immediately following the first three words.

Reply Link Service Sequence:

Service Reject (LS_RJT)
signifies rejection of the ABTX command (see 21.5.2)

Accept (ACC)
signifies that the destination N_Port has terminated the Exchange.

— Accept Payload

The format of the Accept Payload is shown in table 63.

Table 63 - ABTX Accept Payload	
Item	Size -Bytes
hex '02000000'	4

21.4.3 Advise Credit (ADVC)

The ADVC Link Service request shall be used to advise the destination N_Port of the estimated end-to-end Credit which the source N_Port requests to be allocated. The ADVC Link Service request shall be a separate Sequence. It may also be requested in a separate Exchange. See 23.8.2.3 for the usage of this frame. The ADVC request may also be used independently from the Estimate Credit procedure (see 23.8).

Protocol:

Advise Credit request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID field designates the source N_Port requesting Credit revision. The D_ID field designates the destination N_Port.

Payload: The format of the Payload is shown in table 64. The Payload shall contain estimated Credit (M+1) in the end-to-end Credit field of the appropriate Class Service Parameters (see 23.6) as indicated by the Class Validity bit. The Class Validity bit = 1 for each Class Service Parameters of the ADVC Payload identifies the Class for which a revised end-to-end Credit is requested. The other Service Parameter fields shall be ignored by the receiver.

Table 64 - ADVC Payload	
Item	Size -Bytes
hex '0D000000'	4
Common Service Parameters	16
N_Port_Name	8
Node_Name	8
Class 1 Service Parameters	16
Class 2 Service Parameters	16
Class 3 Service Parameters	16
Reserved	16
Vendor Version Level	16

Reply Link Service Sequence:

Service Reject (LS_RJT)
signifies rejection of the ADVC command (see 21.5.2)

Accept (ACC)
signifies successful completion of the ADVC function and permanently replaces the end-to-end Credit in effect for the current N_Port Login.

— Accept Payload
The format of the Accept Payload is shown in table 65. The Payload shall contain the revised end-to-end Credit allocated in the Credit field for the appropriate Class Service Parameters as indicated by the Class Validity bit. The revised end-to-end Credit shall replace the end-to-end Credit for the current Login for the N_Port transmitting the Accept Sequence (see 23.6.8.7). The Class Validity bit = 1 for each Class Service Parameter group of the ADVC Payload identifies each Class for which the end-to-end Credit is updated or revised. The other Service Parameter fields shall be ignored by the receiver. This revised end-to-end Credit value is determined by the destination N_Port based on its buffering scheme, buffer management, buffer availability, and N_Port processing time. See 23.8.2.3 for the determination of this value.

Table 65 - ADVC Accept Payload	
Item	Size -Bytes
hex '02000000'	4
Common Service Parameters	16
N_Port_Name	8
Node_Name	8
Class 1 Service Parameters	16
Class 2 Service Parameters	16
Class 3 Service Parameters	16
Reserved	16
Vendor Version Level	16

21.4.4 Echo (ECHO)

The Echo Extended Link Service request Sequence shall consist of a single frame requesting the Recipient to transmit the Payload contents, following the LS_Command, back to the Initiator of the Echo command in the same order as received using the ACC reply Sequence consisting of a singe frame. The Echo frame shall indicate End_Sequence and Sequence Initiative transfer as well as other appropriate F_CTL bits. The Echo Extended Link Service request provides a means to transmit a Data frame and have the Payload content returned for a simple loop-back diagnostic function. The Echo command shall be transmitted as a one frame Sequence and the ACC reply Sequence is also a one frame Sequence. The Echo Protocol shall be transmitted as an Exchange which is separate from any other Exchange. The Echo Protocol is applicable to Class 1, 2, and 3.

Protocol:

Echo request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The D_ID field designates the destination of the request while the S_ID field designates the source of the request.

Payload: The format of the Payload is shown in table 66.

Table 66 - ECHO Payload	
Item	Size -Bytes
hex '10000000'	4
ECHO data	max frame

The Payload size is limited by the smallest Data Field size supported by the destination N_Port, the Fabric, and the source destination N_Port for the Class of Service being used since the Accept frame shall be equal in size to the Echo Data Field size.

Reply Sequence:

Service Reject (LS_RJT)
signifies rejection of the ECHO command (see 21.5.2)

Accept (ACC)
signifies successful completion of the ECHO function.

– Accept Payload
The format of the Accept Payload is shown in table 67. The Payload shall contain the ECHO data contained in the Payload of the ECHO request frame.

Table 67 - ECHO Accept Payload	
Item	Size -Bytes
hex '02000000'	4
ECHO data	= Echo

21.4.5 Estimate Credit (ESTC)

The ESTC Link Service request shall be used to estimate the minimum Credit required to achieve the maximum bandwidth for a given distance between an N_Port pair. The ESTC Link Service request shall have the frame size as determined by Login with the destination N_Port.

The Class of the **SOF** delimiter of the ESTC request identifies the Class for which Credit is being estimated. The destination N_Port shall acknowledge Data frames as specified by its Login parameters. See 23.8.2.2 for the usage of this frame.

Protocol:

Estimate Credit request Sequence
No reply Sequence

Format: FT_1
Addressing: The S_ID field designates the source N_Port requesting the Credit estimate. The D_ID field designates the destination N_Port specified in the Establish Streaming frame.
Payload: The format of the Payload is shown in table 68. The first word of the Payload of the first frame of the Sequence shall contain the LS_Command code. The remainder of the Payload shall be the size as determined by Login. The content of the Payload after the LS_Command and for subsequent frames shall be valid data bytes.

Table 68 - ESTC Payload

Item	Size -Bytes
hex '0C000000'	4
Any data	max

Reply Link Service Sequence:

None.

21.4.6 Establish Streaming (ESTS)

The ESTS Link Service request Sequence requests a temporary allocation of Credit known as Streaming Credit large enough to perform continuous streaming of Data frames. The **SOF** delimiter of the ESTC request identifies the Class for which Credit is being estimated. See 23.8.2.1 for the usage of this frame.
Protocol:

Establish Streaming request Sequence
Accept reply Sequence

Format: FT_1
Addressing: The S_ID field designates the source N_Port requesting Streaming. The D_ID field designates the destination N_Port addressed.
Payload: The format of the Payload is shown in table 69.

Table 69 - ESTS Payload

Item	Size -Bytes
hex '0B000000'	4

Reply Link Service Sequence:

Service Reject (LS_RJT)
signifies rejection of the ESTS command (see 21.5.2)
Accept (ACC)
signifies successful completion of the ESTS function.
— Accept Payload
The format of the Accept Payload is shown in table 70. The Payload shall contain Streaming Credit (L) allocated in the Credit field of the appropriate Class Service Parameters. The Class Validity bit = 1 identifies the Class which contains the Streaming Credit (L). The other Service Parameter fields shall be ignored by the receiver.

Table 70 - ESTS Accept Payload

Item	Size -Bytes
hex '02000000'	4
Common Service Parameters	16
N_Port_Name	8
Node_Name	8
Class 1 Service Parameters	16
Class 2 Service Parameters	16
Class 3 Service Parameters	16
Reserved	16
Vendor Version Level	16

21.4.7 Login (FLOGI/PLOGI)

The Login Link Service request shall transfer Service Parameters from the initiating N_Port to the N_Port or F_Port associated with the Destination Identifier. The PLOGI frame provides the means by which an N_Port may request Login with another N_Port prior to other Data frame transfers. The FLOGI frame provides the m ans by which an N_Port may request Login with the Fabric (see 23.1). The term LOGI shall be considered to be equivalent to PLOGI or FLOGI.

The interchange of Login information shall establish the operating environment between the two N_Ports or between an N_Port and F_Port. Three Classes of Service are available depending on the support provided by the Fabric, if present. See 23.6 for a definition of Service Parameters.

In order to Login with the Fabric and determine the Fabric operating characteristics, an N_Port shall specify the Destination Identifier as the well-known F_Port Identifier (i.e., hex 'FFFFFE').

In order to direct the Login Link Service frame to a Fibre Channel Service, an N_Port shall specify the appropriate well-known Address Identifier (see table 33 in 18.3).

When an N_Port receives a Login from an N_Port, all Active or Open Sequences with the N_Port performing reLogin shall be abnormally terminated.

Protocol:

Login request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID field designates the source N_Port requesting Login. If unidentified, as in Fabric Login, binary zeros are used. The D_ID field designates the destination N_Port or F_Port of the Login.

Payload: The format of the Payload is shown in table 71.

Table 71 - LOGI Payload

Item	Size -Bytes
hex '03000000' for PLOGI hex '04000000' for FLOGI	4
Common Service Parameters	16
N_Port_Name	8
Node_Name	8
Class 1 Service Parameters	16
Class 2 Service Parameters	16
Class 3 Service Parameters	16
Reserved	16
Vendor Version Level	16

Service Parameters are d fined in 23.6.

Reply Link Service Sequence:

Service Reject (LS_RJT)
signifies rejection of the LOGI command (see 21.5.2)

Accept (ACC)
signifies successful completion of the LOGI command

– Accept Payload
The format of the Accept Payload is shown in table 72 (see 23.6).

Table 72 - LOGI Accept Payload

Item	Size -Bytes
hex '02000000'	4
Common Service Parameters	16
Port_Name	8
Node_Name or Fabric_Name	8
Class 1 Service Parameters	16
Class 2 Service Parameters	16
Class 3 Service Parameters	16
Reserved	16
Vendor Version Level	16

21.4.8 Logout (LOGO)

The Logout Link Service request shall request invalidation of the Service Parameters and Port_Name which have been saved by an N_Port, freeing those resources. This provides a means by which an N_Port may request Logout or remove service between two N_Ports. (see 23.5)

The N_Port Identifier and Port_Name of the N_Port requesting Logout is identified in the Payload. This allows an N_Port to Logout its old Identifier using a new Identifier after its native N_Port Identifier has changed. Both the Source N_Port and the Destination N_Port of the Logout request Sequence shall abnormally terminate all Open Exchanges which used the N_Port identifier indicated in the Payload of the Logout request Sequence.

Protocol:

Logout request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The S_ID field designates the source N_Port requesting Logout. The D_ID field designates the destination N_Port of the Logout request.
Payload: The format of the Payload is shown in table 73.

Table 73 - LOGO Payload	
Item	Size -Bytes
hex '05000000'	4
Reserved	1
N_Port Identifier	3
Port_Name	8

Reply Link Service Sequence:

Service Reject (LS_RJT)
signifies rejection of the LOGO command (see 21.5.2)
Accept (ACC)
signifies that Service has been removed.
— Accept Payload:
The format of the Accept Payload is shown in table 74.

Table 74 - LOGO Accept Payload	
Item	Size -Bytes
hex '02000000'	4

21.4.9 Read Connection Status (RCS)

The RCS Link Service request Sequence requests the Fabric Controller to return the current Dedicated Connection status for the N_Port specified in the Payload of the RCS frame. The RCS request provides the means by which an N_Port may interrogate the Fabric for the Connection status of other N_Ports within the Fabric.

In order to direct the RCS Link Service request frame to the Fabric, an N_Port specifies the Destination Identifier as a well-known Fabric Controller address (i.e., hex 'FFFFFD').
Protocol:

Read Connection Status request Sequence
Accept (ACC) reply Sequence

F rmat: FT_1

Addressing: The S_ID field designates the sourc N_Port requesting Connection status. The D_ID field is the Fabric Controller, hex 'FFFFFD'.
Payload: The format of the Payload is shown in table 75. The first word of the Payload shall contain the LS_Command code. The second word shall contain the N_Port address identifier for which Connection status is being requested.

Table 75 - RCS Payload	
Item	Size -Bytes
hex '07000000'	4
reserved	1
N_Port Identifier	3

Reply Link Service Sequence:

Service Reject (LS_RJT)
signifies rejection of the RCS command (see 21.5.2)
Accept (ACC)
signifies that the Fabric has completed the request.
— Accept Payload
The format of the Accept Payload is shown in table 76.

Table 76 - RCS Accept Payload	
Item	Size -Bytes
hex '02000000'	4
Connection Status (Word 1, bits 31-24)	1
N_Port Identifier (Word 1, bits 23-0)	3

Connection Status Codes:

Bit 31 - Connect-request delivered

If bit 31 is zero, the specified N_Port is either not Connected, or is involved in an Established Connection based on the setting of bit 29. If bit 31 is one, a connect-request has been delivered to the specified N_Port, but the N_Port has not yet responded with a proper response frame and a Dedicated Connection does not yet exist.

Bit 30 - Connect-request stacked

If bit 30 is zero, no connect-request is stacked for the specified N_Port on behalf of the requesting N_Port. If bit 30 is one, one or more connect-requests are stacked, but have not been

delivered to the specified N_Port on behalf of the requesting N_Port.

Bit 29 - Connection established

If bit 29 is zero, the specified N_Port in the RCS request is not in a Dedicated Connection. If bit 29 is one, the specified N_Port is involved in a Dedicated Connection. When bit 29 is one, the address identifier in bits 23-0 identifies the other N_Port involved in the Dedicated Connection.

Bit 28 - Intermix mode

If bit 28 is zero, the N_Port specified in the RCS frame is not functioning in Intermix mode. If bit 28 is one, the N_Port specified in the request is functioning in intermix mode. An N_Port is functioning in intermix mode if both the N_Port and the F_Port have both previously indicated that each supports intermix during Login. See 23.6 and 22.4 for more discussion of intermix.

Bits 23 - 0

Bits 23 through 0 specify the address identifier of the N_Port involved in a Dedicated Connection with the other N_Port specified by the RCS Link Service request Sequence frame (i.e., bit 29 = 1).

21.4.10 Read Exchange Status Block (RES)

The RES Link Service request Sequence requests an N_Port to return the Exchange Status Block for the RX_ID or OX_ID originated by the S_ID specified in the Payload of this request Sequence. The specification of OX_ID and RX_ID may be useful or required information for the destination N_Port to locate the status information requested. A Responder destination N_Port would use RX_ID and ignore the OX_ID. Similarly, the Originator would use the OX_ID and ignore the RX_ID. This provides the N_Port transmitting the request with information regarding the current status of the Exchange specified.

If the destination N_Port of the RES request determines that the SEQ_ID, Originator S_ID, OX_ID, or RX_ID are inconsistent, then it shall reply with an LS_RJT Sequence with a reason code that it is unable to perform the command request.

Protocol:

Read Exchang Status request Sequence
Accept (ACC) reply Sequence

Format: FT_1
Addressing: The S_ID field designates the source N_Port requesting the Exchange Status Block. The D_ID field designates the destination N_Port to which the request is being made.
Payload: The format of the Payload is shown in table 77. The Payload shall include an Association Header for the Exchange if the destination N_Port requires X_ID reassignment.

Table 77 - RES Payload

Item	Size -Bytes
hex '08000000'	4
reserved	1
Originator S_ID	3
OX_ID	2
RX_ID	2
Association Header (optionally required)	32

Reply Link Service Sequence:

Service Reject (LS_RJT)
signifies rejection of the RES command (see 21.5.2)
Accept
signifies that the N_Port has transmitted the requested data.
— Accept Payload:
The format of the Accept Payload is shown in table 78. The format of the Exchange Status Block is specified in 24.8.1.

Table 78 - RES Accept Payload

Item	Size -Bytes
hex '02000000'	4
Exchange Status Block (see 24.8.1)	N
Association Header (optionally required)	32

21.4.11 Read Link Error Status Block (RLS)

The RLS Link Service request Sequence requests an N_Port or F_Port to return the Link Error Status Block associated with the Port Identifier specified in the Payload of this frame. This provides the N_Port transmitting the request with information regarding Link Errors detected within the destination N_Port or F_Port. If an F_Port does not support the Link Error Status Block, it shall reply with an LS_RJT with a reason code of unable to perform command request.

Protocol:

Read Link Error Status request Sequence
Accept (ACC) reply Sequence

Format: FT_1
Addressing: The S_ID field designates the source N_Port requesting the Link Error Status Block. The D_ID field designates the destination N_Port or F_Port (FFFFFE) to which the request is being made.
Payload: The format of the Payload is shown in table 79.

Table 79 - RLS Payload

Item	Size -Bytes
hex '0F000000'	4
reserved	1
Port Identifier	3

Reply Link Service Sequence:

Service Reject (LS_RJT)
  signifies rejection of the RLS command (see 21.5.2)
Accept
  signifies that the N_Port has transmitted the requested data.
– Accept Payload:
  The format of the Accept Payload is shown in table 80. The format of the Link Error Status Block is specified in 29.8.

Table 80 - RLS Accept Payload

Item	Size -Bytes
hex '02000000'	4
Link Error Status Block (see 29.8)	24

21.4.12 Read Sequence Status Block (RSS)

The RSS Link Service request Sequence requests an N_Port to return the Sequence Status Block for the SEQ_ID specified in the Payload of this frame. The Payload also specifies the S_ID of the Exchange Originator as well as the associated OX_ID and RX_ID. The specification of OX_ID and RX_ID may be useful or required information for the destination N_Port to locate the status information requested. A Responder destination N_Port may use RX_ID and ignore the OX_ID. Similarly, the Originator would use the OX_ID and ignore the RX_ID. This provides the N_Port transmitting the request with information regarding the current status of the Sequence it identified.

If the destination N_Port of the RSS request determines that the SEQ_ID, Originator S_ID, OX_ID, or RX_ID are inconsistent then it shall reply with an LS_RJT Sequence with a reason code that it is unable to perform the command request.

Protocol:

Read Sequence Status request Sequence
Accept (ACC) reply Sequence

Format: FT_1
Addressing: The S_ID field designates the source N_Port requesting the Sequence Status Block. The D_ID field designates the destination N_Port to which the request is being made.
Payload: The format of the Payload is shown in table 81.

Table 81 - RSS Payload

Item	Size -Bytes
hex '09000000'	4
SEQ_ID	1
Originator S_ID	3
OX_ID	2
RX_ID	2

Reply Link S rvice Sequenc :

Service Reject (LS_RJT)
signifies rejection of the RSS command (see 21.5.2)

Accept
signifies that the N_Port has transmitted the requested data.

— Accept Payload:
The format of the Accept Payload is shown in table 82. The format of the Sequence Status Block is specified in 24.8.2.

Table 82 - RSS Accept Payload

Item	Size -Bytes
hex '02000000'	4
Sequence Status Block (see 24.8.2)	N

21.4.13 Read Timeout Value (RTV)

The RTV Link Service request Sequence requests an N_Port or F_Port (hex 'FFFFFE') to return the Resource_Allocation_Timeout Value (R_A_TOV) and the Error_Detect_Timeout Value (E_D_TOV) in the Accept reply Sequence. This provides the N_Port transmitting the request with information regarding these values from another N_Port or from the F_Port. Usage of E_D_TOV and R_A_TOV is discussed in 29.2.1.

Protocol:

Read Timeout Value (RTV) request Sequence
Accept (ACC) reply Sequence

Format: FT_1
Addressing: The S_ID field designates the source N_Port requesting the timeout interval values. The D_ID field designates the destination N_Port or F_Port to which the request is being made.
Payload: The format of the Payload is shown in table 83.

Table 83 - RTV Payload

Item	Size -Bytes
hex '0E000000'	4

Reply Link Service Sequenc :

Service Reject (LS_RJT)
signifies rejection of the RTV command (see 21.5.2)

Accept
signifies that the N_Port or F_Port has transmitted the requested data.

— Accept Payload:
The format of the Accept Payload is shown in table 84. Timeout values are specified as a count of 1 millisecond increments. Therefore, a value of hex '0000000A' specifies a time period of 10 milliseconds.

Table 84 - RTV Accept Payload

Item	Size -Bytes
hex '02000000'	4
Resource_Allocation_Timeout Value (R_A_TOV) (see 29.2.1)	4
Error_Detect_Timeout Value (E_D_TOV) (see 29.2.1)	4

21.4.14 Reinstate Recovery Qualifier (RRQ)

The Reinstate Recovery Qualifier Link Service request shall be used to notify the destination N_Port that the Recovery_Qualifier shall be available for reuse. The Recovery_Qualifier (S_ID, D_ID, OX_ID, RX_ID, and low SEQ_CNT - high SEQ_CNT) shall be associated with an Exchange in which the Abort Sequence or Abort Exchange was previously performed.

In the case of Abort Exchange, the ESB and Recovery_Qualifier are immediately available for reuse. In the case of Abort Sequence Protocol, the Recovery_Qualifier is purged. A request to Reinstate the Recovery_Qualifier shall only be accepted if the request is made by the Originator N_Port or the Responder N_Port of the target Exchange.

A separate Exchange shall be used to reinstate the Recovery_Qualifier. The Payload shall contain the OX_ID and RX_ID for the Exchange Recovery_Qualifier, as well as the S_ID of the N_Port which originated the Exchange. Resources associated with the OX_ID in the Originator, and with the RX_ID in the Responder, shall be released following transmission and

reception of the Accept reply Sequence if the Exchange had been aborted with ABTX.

Both the Originator and Responder shall ensure that the OX_ID and RX_ID pair being terminated are currently associated with the OX_ID and RX_ID pair specified in the RRQ request.

The Recovery_Qualifier range shall be timed out for an R_A_TOV timeout period (i.e., RRQ shall not be transmitted until an R_A_TOV timeout period after BA_ACC for ABTS or ACC for ABTX has been received) by the N_Port which transmitted and successfully completed either the ABTX or the ABTS frame.

The addressing is fully specified by the D_ID and S_ID fields.

Protocol:

Reinstate Recovery_Qualifier request Sequence

Accept reply Sequence

Format: FT_1

Addressing: The D_ID field designates the destination N_Port of the RRQ request Sequence while the S_ID field designates the source N_Port which is requesting that the Recovery_Qualifier be reinstated.

X_ID: A separate and distinct Exchange is required.

SEQ_ID, SEQ_CNT: The SEQ_ID and the SEQ_CNT shall be appropriate for an Active Sequence.

Payload: The format of the Payload is shown in table 85.

Table 85 - RRQ Payload

Item	Size -Bytes
hex '12000000'	4
reserved	1
Originator S_ID	3
OX_ID	2
RX_ID	2
Association Header (optionally required)	32

If the Sequence Recipient has indicated that X_ID reassignment is required during Login, the Sequence Initiator shall include the Association Header in the Payload of the RRQ requ st associated with the Exchange for which the Recovery_Qualifier is being reinstated.

Reply Link Service Sequence:

Service Reject (LS_RJT)
signifies rejection of the RRQ command (see 21.5.2)

Accept (ACC)
signifies that the destination N_Port reinstated the Recovery_Qualifier.

– Accept Payload
The format of the Accept Payload is shown in table 86.

Table 86 - RRQ Accept Payload

Item	Size -Bytes
hex '02000000'	4

21.4.15 Request Sequence Initiative (RSI)

The Request Sequence Initiative Link Service request shall be used to request that Sequence Initiative be passed to the Sequence Recipient of an Exchange in progress. A request to pass Sequence Initiative shall only be accepted if the request is made by the Originator N_Port or the Responder N_Port of the target Exchange. A separate Exchange shall be used to perform the Request Sequence Initiative. The Payload shall contain the OX_ID and RX_ID for the target Exchange, as well as the S_ID of the N_Port which originated the Exchange. The Accept reply is sent subsequent to the transfer of Sequence Initiative on the target Exchange.

Transmission of RSI is allowed while the identified Exchange is Open. Both the Originator and Responder shall ensure that the OX_ID and RX_ID pair for which Sequence Initiative is being passed are currently associated with the OX_ID and RX_ID pair specified in the RSI request.

If there are any Sequences Active for the target Exchange, the Sequence Initiator of the Active Sequence of the target Exchange shall terminate them and transfer Sequence Initiative as follows:

- If there is an Activ Sequ nce for which the last Data frame has not been transmitted, the Sequence Initiator of the target Exchange shall terminate the Sequence by transmitting a Data frame with the End_Sequence and Sequence Initiative bits set to 1.

— If there are no Data frames to be sent for the Active Sequence, the Sequence Initiator of the target Exchange shall transmit a NOP Basic Link Service frame with the End_Sequence and Sequence Initiative bits set to 1 in F_CTL.

If there are no Sequences Active, the Sequence Initiator of the target Exchange shall transfer Sequence Initiative by initiating a new Sequence consisting of a single NOP Basic Link Service frame (a one frame Sequence) with the End_Sequence and Sequence Initiative bits set to 1 in F_CTL.

The Accept to the Exchange requesting Sequence Initiative shall be transmitted after Sequence Initiative has been passed (see 24.6.4) on the target Exchange.

The addressing is fully specified by the D_ID and S_ID fields.

Protocol:

Request Sequence Initiative request Sequence
Accept reply Sequence

Format: FT_1

Addressing: The D_ID field designates the destination N_Port of the Exchange for which Sequence Initiative is being requested and the S_ID field designates the source N_Port which is requesting Sequence Initiative.

X_ID: A separate and distinct Exchange is required other than the Exchange for which Sequence Initiative is being requested in order to properly track status.

SEQ_ID, SEQ_CNT: The SEQ_ID and the SEQ_CNT shall be appropriate for an Active Sequence.

Payload: The format of the Payload is shown in table 87.

Table 87 - RSI Payload

Item	Size -Bytes
hex '0A000000'	4
reserved	1
Originator S_ID	3
OX_ID	2
RX_ID	2
Association Header (optionally required)	32

If the Sequence Recipient has indicated that X_ID reassignment is required during Login, the Sequence Initiator shall include the Association Header in the Payload of the RSI request associated with the Exchange for which the Sequence Initiative is being requested immediately following the first three words.

Reply Link Service Sequence:

Service Reject (LS_RJT)
signifies rejection of the RSI command (see 21.5.2)

Accept (ACC)
signifies that the destination N_Port has transferred the Sequence Initiative for the target Exchange.

— Accept Payload
The format of the Accept Payload is shown in table 88.

Table 88 - RSI Accept Payload

Item	Size -Bytes
hex '02000000'	4

21.4.16 Test (TEST)

The Test Extended Link Service request Sequence shall consist of a single Sequence being transmitted from the Sequence Initiator to the Sequence Recipient. The Test request may be used in diagnostic or testing procedures to provide system loading. There is no reply Sequence. The Payload may consist of any frame size up to the maximum allowable for the Class and other normal Sequence and frame limitations. The Test Link Service request is applicable to Class 1, 2, and 3.

Protocol:

Test request Sequence

Format: FT_1

Addressing: The D_ID field designates the destination of the request while the S_ID field designates the source of the request.

Payload: The format of the Payload is shown in table 89.

Table 89 - TEST Payload	
Item	Size -Bytes
hex '11000000'	4
TEST data	max frame

The Payload size is limited by the smallest Data Field size supported by the destination N_Port and the Fabric for the Class of Service being used.

Reply Sequence:

none

21.5 Extended Link Service reply Sequences

An Extended Link Service reply Sequence shall signify that the Extended Link Service request Sequence is completed. The reply Sequence may contain data in the Payload following the LS_Command code word. The format and meaning of the Payload is specified in the request Extended Link Service definition.

21.5.1 Accept (ACC)

The Accept Link Service reply Sequence shall notify the transmitter of an Link Service request that the Extended Link Service request Sequence has been completed. The Responder shall terminate the Exchange by setting the Last Sequence bit (Bit 20) in F_CTL on the last Data frame of the reply Sequence. The first word of the Payload shall contain hex '02000000'. The remainder of the Payload is unique to the Link Service request.

Protocol:

Accept is the reply Sequence to Login, Logout, Abort Exchange, Read Connection Status, Read Exchange Status Block, Read Link Error Status, Read Timeout Value, Read Sequence Status Block, Reinstate Recovery Qualifier, Request Sequence Initiative, Establish Streaming, and Advise Credit request Sequences.

Format: FT_1

Addressing: The D_ID field designates the source of the Link Service Sequence being accepted while the S_ID field designates the destination of the request Sequence being accepted.

Payload: The Payload content following the LS_Command code (hex '02000000') is defined within individual Link Service requests.

Reply Link Service Sequence:

none

21.5.2 Link Service Reject (LS_RJT)

The Link Service Reject (LS_RJT) shall notify the transmitter of a Link Service request that the Link Service request Sequence has been rejected. A four-byte reason code shall be contained in the Data_Field. Link Service Reject may be transmitted for a variety of conditions which may be unique to a specific Link Service request.

For example, if the Service Parameters specified in a Login frame were logically inconsistent or in error, a P_RJT frame would not be transmitted in response, but rather a Link Service Reject.

Protocol:

LS_RJT may be a reply Sequence to any Extended Link Service request excluding ESTC.

Format: FT_1

Addressing: The D_ID field designates the source of the Link Service request being rejected while the S_ID field designates the destination of the request Data frame Sequence being rejected.

Payload: The first word of the Payload shall contain the LS_Command code (hex '01000000'). The next four bytes of this field shall indicate the reason for rejecting the request (see figure 58 and tables 90 and 91).

```
Service Reject data definition - second word

             First    Second    Third    Fourth
             Byte     Byte      Byte     Byte
          +---------+---------+---------+---------+
Bits      |3       2|2     111|1        |         |
          |1       4|3     876|5       8|7       0|
          +---------+---------+---------+---------+
          |rrrr rrrr|CCCC CCCC|EEEE EEEE|VVVV VVVV|
          +---------+---------+---------+---------+
           L___ ___J L___ ___J L___ ___J L___ ___J
               |         |         |         |
           Reserved   Reason    Reason    Vendor
                      Code    Explanation Unique
```

Figure 58 - LS_RJT format

Table 90 - LS_RJT reason codes

Encoded Value (Bits 23-16)	Description
0000 0001	Invalid LA_Command code
0000 0011	Logical error
0000 0101	Logical busy
0000 0111	Protocol error
0000 1001	Unable to perform command request
0000 1011	Command not supported
Others	Reserved
1111 1111	Vendor Unique Error (See Bits 7-0)

The first error condition encountered shall be the error reported.

- **Bits 23-16 Description**

Invalid LS_Command code

The LS_Command code in the Sequence being rejected is invalid or not supported.

Logical error

The request identified by the LS_Command code and Payload content is invalid or logically inconsistent for the conditions present.

Logical busy

The Link Service is logically busy and unable to process the request at this time.

Protocol Error

This indicates that an error has been detected which violates the rules of the Extended Link Service Protocol which are not specified by other error codes.

Unable to perform command request

The Recipient of a Link Service command is unable to perform the request at this time.

Command not supported

The Recipient of a Link Service command does not support the command requested.

Vendor Unique Error

The Vendor Unique Error bits may be used by Vendors to specify additional reason codes.

- **Bits 15-8 Reason explanation**

Table 91 shows expanded explanations for Link Service commands with the applicable Extended Link Service commands.

Tabl 91 - LS_RJT reason code explanation		
Encoded Value (Bits 15-8)	Description	Applicable commands
0000 0000	No additional explanation	ABTX, ADVC, ESTS, FLOGI, PLOGI, LOGO, RCS, RES, RLS, RSS, RTV, RSI
0000 0001	Service Parm error - Options	FLOGI, PLOGI
0000 0011	Service Parm error - Initiator Ctl	FLOGI, PLOGI
0000 0101	Service Parm error - Recipient Ctl	FLOGI, PLOGI
0000 0111	Service Parm error - Rec Data Field Size	FLOGI, PLOGI
0000 1001	Service Parm error - Concurrent Seq	FLOGI, PLOGI
0000 1011	Service Parm error - Credit	ADVC, FLOGI, PLOGI
0000 1101	Invalid N_Port/F_Port Name	FLOGI, PLOGI
0000 1110	Invalid Node/Fabric Name	FLOGI, PLOGI
0000 1111	Invalid Common Service Parameters	FLOGI, PLOGI
0001 0001	Invalid Association Header	ABTX, RES, RRQ, RSI
0001 0011	Association Header required	ABTX, RES, RRQ, RSI
0001 0101	Invalid Originator S_ID	ABTX, RES, RRQ, RSI, RSS
0001 0111	Invalid OX_ID-RX_ID combination	ABTX, RES, RRQ, RSI, RSS
0001 1001	Command (request) already in progress	ABTX, PLOGI, RSI
0001 1111	Invalid N_Port Identifier	RCS, RLS
0010 0001	Invalid SEQ_ID	RSS
0010 0011	Attempt to abort invalid Exchange	ATBX
0010 0101	Attempt to abort inactive Exchange	ABTX
0010 0111	Recovery_Qualifier required	ABTX
0010 1001	Insufficient resources to support Login	FLOGI, PLOGI
0010 1010	Unable to supply requested Data	ADVC, ESTS, RCS, RES, RLS, RSS, RTV
0010 1100	Request not supported	ADVC, ESTS, ESTC
Others	Reserved	

Reply Link Service Sequence:

none

21.6 FC-4 Link Service

An FC-4 Link Service request solicits a destination Port (F_Port or N_Port) to perform a function or service in order to support an individual FC-4 Device_Data protocol. The Information Category for a request shall be specified as Unsolicited Control. An FC-4 Link Service reply may be transmitted in answer to an FC-4 Link Service request. The Information Category for a reply shall be specified as Solicited Control. Each request or reply shall be composed of a single Sequence. The format of the request or reply shall be specified by the individual FC-4 being supported and is outside the scope of FC-PH. Each Sequence may be composed of one or more frames.

The protocols supported by the FC-4 Link Services shall be performed within a single Exchange, intended exclusively for the purpose. FC-4 Link Service protocols are performed using a two Sequence Exchange. The protocols consist of a request Sequence by the Originator (N_Port), transfer of Sequence Initiative, and a reply Sequence from the Responder (N_Port or F_Port). The Sequence Initiator and Sequence Recipient shall follow the rules for Sequence management and Recovery_Qualifier reuse as specified in clause 24. The following rules regarding Sequence and Exchange management

apply to FC-4 Link Services in addition to the rules specified in clause 24:

- FC-4 Link Services shall only be Exchanges originated following N_Port Login.
- the Originator of the Exchange shall use the Discard multiple Sequences Exchange Error Policy for all FC-4 Link Service Exchanges.
- the Originator of an FC-4 Link Service Exchange shall detect an Exchange error following Sequence Initiative transfer if the reply Sequence is not received within a timeout interval equal to twice the value of R_A_TOV.
- if the Exchange Originator of an FC-4 Link Service Exchange detects an Exchange error, it shall abort the Exchange (ABTX) and retry the protocol of the aborted Exchange with a different Exchange.
- if the Sequence Initiator aborts a Sequence using ABTS (Abort Sequence Protocol) due to receiving an ACK with the Abort Sequence bits (5-4) set to 0 1, the Sequence Initiator shall retry the Sequence after the Basic Accept is received for the aborted Sequence one time only.

21.6.1 Routing control

R_CTL bits 31-28 (Word 0) are set = to 0011 to indicate an FC-4 Link_Data frame. The TYPE field for each FC-4 Link Service frame shall match the FC-4 Device_Data TYPE field as specified in table 36

21.7 Basic Link Service summary

Table 92 summarizes the Basic Link Service request and reply frames. The Payload content and size in bytes are specified. The Payload for the BA_RJT is 4 bytes in size.

Table 92 - Basic Link Service Payload

REQUEST SEQUENCE		REPLY SEQUENCE	
Request	**Payload**	**Reply**	**Accept Payload**
Abort Sequence (ABTS)	none	BA_ACC or BA_RJT	SEQ_ID Validity, SEQ_ID, OX_ID, RX_ID lo SEQ_CNT, hi SEQ_CNT (12 bytes)
No Operation (NOP)	None	None	
Remove Connection (RMC)	None	None	

21.8 Extended Link Service summary

Table 93 summarizes the Extended Link Service request and reply Sequences. The Payload content and size in bytes are specified. The Payload for the LS_RJT is 8 bytes in size.

Table 93 (Page 1 of 2) - Extended Link Service Payload

REQUEST SEQUENCE Request	 Payload	REPLY SEQUENCE Reply	 Accept Payload
Abort Exchange (ABTX)	LS_Command code, Recovery_Qualifier Status S_ID, OX_ID , RX_ID (12 bytes) Association Header (optional, 32 bytes)	Accept (ACC) or LS_RJT	LS_Command code (4 bytes)
Advise Credit (ADVC)	LS_Command code, Common Parameters, Port_Name, Node_Name, Service Parameters (Class 1, 2, 3), Vendor Version (116 bytes)	Accept (ACC) or LS_RJT	LS_Command code, Common Parameters, Port_Name, Node_Name, Service Parameters (Class 1, 2, 3), Vendor Version (116 bytes)
Echo (ECHO)	LS_Command code, up to Max frame (N bytes)	Accept (ACC) or LS_RJT	LS_Command code, Same as Echo (N bytes)
Establish Streaming (ESTS)	LS_Command code (4 bytes)	Accept (ACC) or LS_RJT	LS_Command code, Common Parameters, Port_Name, Node_Name, Service Parameters (Class 1, 2, 3), Vendor Version (116 bytes)
Estimate Credit (ESTC)	LS_Command code, Data (maximum size)	None	
Login (FLOGI/PLOGI)	LS_Command code, Common Parameters, Port_Name, Node_Name, Service Parameters (Class 1, 2, 3), Vendor Version (116 bytes)	Accept (ACC) or LS_RJT	LS_Command code, Common Parameters, Port_Name, Node_Name, Service Parameters (Class 1, 2, 3), Vendor Version (116 bytes)
Logout (LOGO)	LS_Command code Reserved (1), N_Port Identifier (3) Port_Name (8) (16 bytes)	Accept (ACC) or LS_RJT	LS_Command code (4 bytes)
Read Connection Status (RCS)	LS_Command code, Address_Identifier (8 bytes)	Accept (ACC) or LS_RJT	LS_Command code, Status, Address_Identifier (8 bytes)

Tabl 93 (Page 2 of 2) - Extended Link Service Payl ad			
REQUEST SEQUENCE		**REPLY SEQUENCE**	
Request	**Payload**	**Reply**	**Accept Payload**
Read Exchange Status Block (RES)	LS_Command code, Originating S_ID, OX_ID , RX_ID (12 bytes) Association Header (optional, 32 bytes)	Accept (ACC) or LS_RJT	LS_Command code, Exchange SB Format (N bytes)
Read Link Error Status Block (RLS)	LS_Command code, N_Port Identifier (8 bytes)	Accept (ACC) or LS_RJT	LS_Command code, Link Error SB format (24 bytes)
Read Sequence Status Block (RSS)	LS_Command code, SEQ_ID, Originator S_ID, OX_ID, RX_ID (12 bytes)	Accept (ACC) or LS_RJT	LS_Command code, Sequence SB Format (N bytes)
Read Timeout Value (RTV)	LS_Command code, (4 bytes)	Accept (ACC) or LS_RJT	LS_Command code, R_A_TOV, E_D_TOV, (12 bytes)
Reinstate Recovery_Qualifier (RRQ)	LS_Command code, S_ID, OX_ID , RX_ID (12 bytes) Association Header (optional, 32 bytes)	Accept (ACC) or LS_RJT	LS_Command code (4 bytes)
Request Sequence Initiative (RSI)	LS_Command code, Originating S_ID, OX_ID , RX_ID (12 bytes) Association Header (optional, 32 bytes)	Accept (ACC) or LS_RJT	LS_Command code (4 bytes)
Test (TEST)	LS_Command code, up to Max frame (N bytes)	None	

22 Classes of s rvice

Three Classes of service applicable to a Fabric and an N_Port are specified. These Classes of service are distinguished primarily by the methodology with which the communication circuit is allocated and retained between the communicating N_Ports and by the level of delivery integrity required for an application.

A given Fabric or N_Port may support one or more Classes of service. These Classes of service are:

a) Class 1 - Dedicated Connection

b) Class 2 - Multiplex

c) Class 3 - Datagram

Each Class of service may be supported with any of the communication models. Intermix is specified as an option of Class 1 service.

In all Classes of service, the FC-2 Segmentation and Reassembly function makes available to the receiving ULP, the same image of application data as transmitted by the sending ULP (see clause 27).

In all Classes of service, for each frame received, the Fabric shall

a) deliver only one instance of the frame,
b) issue a F_BSY,
c) issue a F_RJT, or
d) discard the frame without issuing any response.

22.1 Class 1 -- Dedicated Connection

Class 1 is a service which provides Dedicated Connections. A Class 1 Connection is requested by an N_Port with another N_Port. An acknowledgement (ACK) is returned by the other N_Port to establish the Connection (see clause 28). In general, once a Connection is established, it shall be retained and guaranteed by the Fabric, until one of these N_Ports requests removal of the Connection. See 16.4 for special conditions in which a Connection may be removed.

NOTE - If a Class 1 Connection can be established between N_Ports of unlike speeds, the method and configuration by which this Connection is established will be transparent to FC-PH.

22.1.1 Class 1 function

A Class 1 Connection is requested by an N_Port with another N_Port via transmission of a frame containing a **SOFc1** (Class 1/**SOFc1**) . The Fabric, if present, allocates a circuit between the requesting N_Port and the destination N_Port. The destination N_Port transmits an ACK indicating its acceptance to the requesting N_Port. Upon successful receipt of the ACK at the requesting N_Port, the Connection is established (see 28.4.1). The Fabric retains the allocated circuit between these two N_Ports, until one of these N_Ports requests the Dedicated Connection be removed.

Even if a Fabric is not present, the requesting N_Port and the destination N_Port follow the same protocol.

Class 1 Delimiters as specified in 22.1.3 are used to establish and remove the Dedicated Connection and to initiate and terminate one or more Sequences within the Dedicated Connection.

22.1.2 Class 1 rules

The following rules apply to exclusive Class 1 Connections. See 22.4 for additional rules applicable to Intermix. To provide a Class 1 Connection, the transmitting and receiving N_Ports, and the Fabric shall obey the following rules:

a) Except for some Link Service Protocols (see clause 21), an N_Port requesting a Class 1 Connection is required to have previously logged in with the Fabric and the N_Ports with which it intends to communicate, either implicitly or explicitly. To login explicitly, the requesting N_Port shall use Fabric and N_Port Login protocols (see clause 23). **(Login)**

b) The Fabric is responsible for establishing a Connection at the request of an N_Port and retaining it until one of the communicating N_Ports explicitly requests removal of the Connection or one of the links attached to the N_Ports participates in a primitive sequence protocol (see 16.4). To establish or remove the Dedicated Connection, the requesting

N_Port shall use the Class 1 Delimiters as specified in 22.1.3. **(Connection** through Connection Sub-Fabric)

c) After transmitting a Class 1/**SOFc1** frame, the N_Port requesting the Connection shall not transmit additional frames to the destination N_Port, until it receives from that destination N_Port, an ACK which shall establish the Connection. **(establish Connection)**

d) Once a Connection is established between two N_Ports, each N_Port shall send frames only to the other N_Port in the Connection until the Connection is removed. All these frames shall contain respective S_ID and D_ID for these N_Ports. **(established Connection)**

e) A destination N_Port shall acknowledge delivery of every valid Data frame with an ACK_1 or ACK_N, or the entire Sequence with a single ACK_0 (see 22.1.5). **(Acknowledgement)**

f) The Sequence Initiator shall increment the SEQ_CNT field of each successive frame transmitted within a Sequence. The Fabric shall guarantee delivery of the frames at the receiver in the same order of transmission within the Sequence (see 24.3.6). **(sequential delivery)**

g) Each N_Port of an established Connection may originate multiple Exchanges and initiate multiple Sequences within that Connection. The N_Port originating an Exchange shall assign an X_ID unique to the Originator called OX_ID and the Responder of the Exchange shall assign an X_ID unique to the Responder called RX_ID. Thus within a given Connection, the value of OX_ID or RX_ID is unique to the respective N_Port. The Sequence Initiator shall assign a SEQ_Qualifier, for each Sequence it initiates, which is unique to the Sequence Initiator and the respective Sequence Recipient pair. **(concurrent Exchanges and Sequences)**

h) Communicating N_Ports shall be responsible for end-to-end flow control, without any Fabric involvement. ACK frames are used to perform end-to-end flow control (see 22.1.5). All Class 1 frames except Class 1/**SOFc1**, participate only in end-to-end flow control. A Class 1/**SOFc1** frame participates in both end-to-end and buffer-to-buffer flow controls. **(end-to-end flow control)**

i) The Fabric may reject a request for a Class 1 Connection or issue a busy with a valid reason code. After the Dedicated Connection is established, the Fabric shall not interfere with the Connection. The Fabric shall issue a F_BSY to any Class 2 frame or discard any Class 3 frame, transmitted from a third N_Port and destined to one of the N_Ports engaged in the Connection (see 20.3.3.1 and 20.3.3.3). **(Fabric reject or busy)**

j) The destination N_Port specified in Class 1/**SOFc1** frame may respond with a busy or a reject with a valid reason code to this frame. Once the Dedicated Connection is established, the destination N_Port shall not issue a busy but may issue a reject (see 20.3.3.2 and 20.3.3.3). **(N_Port busy or reject)**

k) The End-to-end Credit established during the Login protocol by interchanging service Parameters shall be honored (see 26.3). At the beginning of a Connection, the End-to-end Credit_Count is reinitialized to the value determined during Login. Class 1/**SOFc1** frame shall share the End-to-end Credit with Class 2 frames (see 26.3). **(Credit)**

l) The Fabric shall guarantee full bandwidth availability to the connected N_Ports (see clause 3 and annex M). **(bandwidth)**

m) Frames within a Sequence are tracked on an Sequence_Qualifier and SEQ_CNT basis (see 18.1.1). **(tracking)**

n) An N_Port or F_Port shall be able to recognize **SOF** delimiters for all Classes of service, whether or not all Classes are supported by the Port, and provide appropriate responses for all Classes with appropriate delimiters. **(invalid Class)**

- If an N_Port which supports only Class 1 receives a Class 2 frame, and
  - is not engaged in a Dedicated Connection, the N_Port shall issue a P_RJT with appropriate Class 2 delimiters and obey buffer-to-buffer flow control rules.
  - is engaged in a Dedicated Connection, the N_Port response is unpredictable.
- If an N_Port which supports only Class 1 receives a Class 3 frame, and
  - is not engaged in a Dedicated Connection, the N_Port shall discard the frame and obey the buffer-to-buffer flow control rules.

— is engaged in a Dedicated Connection, the N_Port shall discard th frame and not follow the buffer-to-buffer flow control rules.

— If an F_Port which supports only Class 1 receives a Class 2 frame, and

— is not engaged in a Dedicated Connection, the F_Port shall issue a F_RJT with appropriate Class 2 delimiters and obey buffer-to-buffer flow control rules.

— is engaged in a Dedicated Connection, the F_Port response is unpredictable.

— If an F_Port which supports only Class 1 receives a Class 3 frame, and

— is not engaged in a Dedicated Connection, the F_Port shall discard the frame and obey buffer-to-buffer flow control rules.

— is engaged in a Dedicated Connection, the F_Port response is unpredictable.

o) If an N_Port does not support Class 1 and receives a Class 1/**SOFc1** frame, the N_Port shall issue a P_RJT with a **SOFn1** and **EOFdt** with a reason code of Class not supported. If an F_Port does not support Class 1 and receives a Class 1/**SOFc1** frame, the F_Port shall issue a F_RJT with a **SOFn1** and **EOFdt** with a reason code of Class not supported. **(Class 1 not supported)**

p) If an F_Port, not engaged in a Dedicated Connection, receives a frame with a **SOFi1** or **SOFn1**, the F_Port response is unpredictable. However, the buffer-to-buffer control shall remain unaffected. **(Invalid protocol)**

22.1.3 Class 1 delimiters

A Dedicated Connection is requested by transmitting a Data frame using an **SOFc1** delimiter. **SOFc1** initiates the first Sequence; subsequent Sequences are initiated with an **SOFi1**. All frames other than the first within a Sequence are started by **SOFn1**.

All frames other than the last frame within a Sequence are terminated by **EOFn**. Each Sequence is terminated using an **EOFt** (see 22.1.5). An **EOFdt** contained in a frame terminates the Sequence in which the frame is sent and it also serves to remove the Dedicated Connection. Other open Sequences in progress are also terminated. Exchanges and Sequences may be left in indeterminate state from the perspective of ULPs.

22.1.4 Class 1 frame size

The size of Data_Field of a frame using the **SOFc1** delimiter is limited by the smaller value of the maximum Data_Field size supported for frames with **SOFc1** by the Fabric and the destination N_Port. Subsequent frames, after a Dedicated Connection is established, are limited only by the maximum Data Field size supported by the destination N_Port.

22.1.5 Class 1 flow control

ACK frames are used to perform Class 1 end-to-end flow control. ACK frames are started with **SOFn1**. The ACK used to terminate a Sequence shall end with **EOFt**. The ACK used to terminate the Connection shall end with **EOFdt**. All ACK frames which do not terminate a Sequence shall end with **EOFn**.

All Class 1 frames shall follow end-to-end flow control rules (see 26.4.1). The Class 1/**SOFc1** frame shall follow both end-to-end and buffer-to-buffer flow control rules (see 26.5.1).

22.1.6 Stacked connect-requests

Stacked connect-requests is a feature which may be provided by the Fabric (see 28.5.2).

22.2 Class 2 -- Multiplex

This operating environment provides Connectionless service with notification of non-delivery between two N_Ports. This service allows one N_Port to transmit consecutive frames to multiple destinations without establishing a Dedicated Connection with any specific N_Port. Conversely, this service also allows one N_Port to receive consecutive frames from one or more N_Ports without having establish d Dedicated Connections with any of them.

22.2.1 Class 2 functi n

A Class 2 service is requested by an N_Port on a frame by frame basis. The Fabric, if present, routes the frame to the destination N_Port. If the N_Port transmits consecutive frames to multiple destinations, the Fabric demultiplexes them to the requested destinations.

Class 2 Delimiters are used to indicate the requested service and to initiate and terminate one or more Sequences as described in 22.2.3. Since Class 2 is Connectionless, the question of service removal does not arise.

22.2.2 Class 2 rules

To provide Class 2 service, the transmitting and receiving N_Ports, and the Fabric shall obey the following rules:

a) Except for some Link Service Protocols (see clause 21), an N_Port supporting Class 2 service is required to have logged in with the Fabric and the N_Ports with which it intends to communicate, either explicitly or implicitly. To login explicitly, the requesting N_Port shall use Fabric and N_Port Login protocols (see clause 23). **(Login)**

b) The Fabric routes the frames through Connectionless Sub-Fabric, without establishing a Dedicated Connection between communicating N_Ports. To obtain Class 2 service from the Fabric, the N_Port shall use the Class 2 Delimiters as specified in 22.2.3. **(Connectionless service)**

c) An N_Port is allowed to send consecutive frames to one or more destinations. This enables an N_Port to demultiplex multiple Sequences to a single or multiple destinations concurrently (see 22.2.3). **(demultiplexing)**

d) A given N_Port may receive consecutive frames from different sources. Each source is allowed to send consecutive frames for one or more Sequences. **(multiplexing)**

e) A destination N_Port shall provide an acknowledgement to the source for each valid Data frame received. As in Class 1, the destination N_Port shall use ACK for the acknowledgement (see 22.2.5). If unable to deliver ACK, the Fabric shall return a F_BSY or F_RJT. **(Acknowledgement)**

f) The Sequence Initiator shall increment the SEQ_CNT field of each successive frame transmitted within a Sequence. However, the Fabric may not guarantee delivery to the destination in the same order of transmission (see 24.3.6). **(non-sequential delivery)**

g) An N_Port may originate multiple Exchanges and initiate multiple Sequences with one or more destination N_Ports. The N_Port originating an Exchange shall assign an X_ID unique to the Originator called OX_ID and the Responder of the Exchange shall assign an X_ID unique to the Responder called RX_ID. The value of OX_ID or RX_ID is unique to a given N_Port. The Sequence Initiator shall assign a SEQ_ID, for each Sequence it initiates, which is unique to the Sequence Initiator and the respective Sequence Recipient pair while the Sequence is Open (see 24.5). **(concurrent Exchanges and Sequences)**

h) Each F_Port (the local and the remote) exercises buffer-to-buffer flow control with the N_Port to which it is directly attached. End-to-end flow control is performed by communicating N_Ports. ACK frames are used to perform end-to-end flow control and R_RDY is used for buffer-to-buffer flow control. **(dual flow control)**

i) If the Fabric is unable to deliver the frame to the destination N_Port, then the source is notified of each frame not delivered by an F_BSY or F_RJT frame from the Fabric with corresponding D_ID, S_ID, OX_ID, RX_ID, SEQ_ID, and SEQ_CNT. The source is also notified of valid frames busied or rejected by the destination N_Port by P_BSY or P_RJT. **(non-delivery)**

j) A busy or reject may be issued by an F_Port or the destination N_Port with a valid reason code (see 23.7, 23.6, 20.3.3.3, and 20.3.3.2). **(busy/reject)**

k) If a Class 2 Data frame is busied, the sender shall retransmit the busied frame up to the ability of the sender to retry, including zero. **(retransmit)**

l) The Credit established during the Login protocol by interchanging Service Parameters shall be honored (see 26.3 for more on Credit.). Class 2 may share the Credit for Connectionless service with Class 3 and Class 1/SOFc1 frames (see 26.3). **(Credit)**

m) Effective transfer rate between any given N_Port pair is dependent upon the number of N_Ports a given N_Port is demultiplexing to and multiplexing from. **(bandwidth)**

n) Frames within a Sequence are tracked on a Sequence_Qualifier and SEQ_CNT basis (see 18.1.1). **(tracking)**

o) An N_Port or F_Port shall be able to recognize **SOF** delimiters for all Classes of service, whether or not all Classes are supported by the Port, and provide appropriate responses for all Classes with appropriate delimiters. An N_Port which supports only Class 2 shall issue a P_RJT for Class 1 frames with appropriate Class 1 delimiters and discard Class 3 frames, while obeying the buffer-to-buffer flow control rules in both cases. An F_Port which supports only Class 2 shall issue a F_RJT for Class 1 frames with appropriate Class 1 delimiters and discard Class 3 frames, while obeying the buffer-to-buffer flow control rules in both cases. **(invalid Class)**

22.2.3 Class 2 delimiters

Sequences are initiated by transmitting a frame started by an **SOFi2**. Subsequent frames within a Sequence are started by an **SOFn2**. A Sequence is normally terminated with a frame ended by **EOFt**. All frames other than the last frame within the Sequence are ended with an **EOFn**.

22.2.4 Class 2 frame size

The size of each frame transmitted is limited by the smaller value of the maximum Data Field size supported by the Fabric or by the receiving N_Port. Each frame is routed through the Fabric, if present, as a separate entity.

22.2.5 Class 2 flow control

Class 2 service uses both buffer-to-buffer and end-to-end flow controls. R_RDY (Receiver Ready) is used for buffer to buffer flow control. R_RDY is transmitted by the F_Port to the N_Port originating the Class 2 frame to indicate that a buffer is available for further frame reception at the F_Port. R_RDY is transmitted by the destination N_Port to the attached F_Port to indicate that a buffer is available for further frame reception at the destination N_Port.

ACK frames are used to perform end-to-end flow control. ACK frames shall begin with **SOFn2**. The ACK used to terminate a Sequence shall end with **EOFt**. All ACK frames which do not terminate a Sequence shall end with **EOFn**.

All Class 2 frames shall follow both buffer-to-buffer and end-to-end flow control rules (see 26.5.1 and 26.5.1).

22.3 Class 3 -- Datagram

This operating environment provides Connectionless service without any notification of non-delivery (BSY or RJT), delivery (ACK), or end-to-end flow control between two N_Ports. The Fabric, if present, and the destination N_Port are allowed to discard Class 3 frames without any notification to the transmitting N_Port. This service allows one N_Port to transmit consecutive frames to multiple destinations without establishing a Dedicated Connection with any specific N_Port. Conversely, this service also allows one N_Port to receive consecutive frames from one or more N_Ports without having established Dedicated Connections with any of them.

22.3.1 Class 3 function

A Class 3 service is requested by an N_Port on a frame by frame basis. The Fabric, if present, routes the frame to the destination N_Port. If the N_Port transmits consecutive frames to multiple destinations, the Fabric demultiplexes them to the requested destinations.

Class 3 Delimiters are used to indicate the requested service and to initiate and terminate one or more Sequences as described in 22.3.3. Since Class 3 is Connectionless, the question of service removal does not arise.

22.3.2 Class 3 rules

To provide Class 3 service, the transmitting and receiving N_Ports, and the Fabric shall obey the following rules:

a) Except for some Link Service Protocols (see clause 21), an N_Port supporting Class 3 servic is required to have logged in with the Fabric or the N_Ports, either explicitly or implicitly. To login explicitly, the requesting

N_Port shall use Fabric and N_Port Login protocols (see clause 23). **(Login)**

b) The Fabric routes the frames through Connectionless Sub_Fabric, without establishing a Dedicated Connection between communicating N_Ports. To obtain Class 3 service from the Fabric, the N_Port shall use the Class 3 Delimiters as specified in 22.3.3. **(Connectionless service)**

c) A given N_Port is allowed to send consecutive frames to one or more destinations. This enables an N_Port to demultiplex multiple Sequences to single or multiple destinations concurrently. **(demultiplexing)**

d) A given N_Port may receive consecutive frames from one or more source N_Ports. Each source N_Port is allowed to send consecutive frames for one or more Sequences. **(multiplexing)**

e) A destination N_Port shall not provide acknowledgement (ACK) to the source for any valid frame received. **(absence of acknowledgement)**

f) The Sequence Initiator shall increment the SEQ_CNT field of each successive frame transmitted within a Sequence. However, the Fabric may not guarantee delivery at the receiver in the same order of transmission (see 24.3.6). **(non-sequential delivery)**

g) An N_Port may originate Exchanges and initiate Sequences with one or more destination N_Ports. The N_Port originating an Exchange shall assign an X_ID unique to the Originator called OX_ID and the Responder of the Exchange shall assign an X_ID unique to the Responder called RX_ID. Responder may assign an RX_ID in the first Sequence it transmits. The value of OX_ID or RX_ID is unique to a given N_Port. The Sequence Initiator shall assign a SEQ_Qualifier, for each Sequence it initiates, which is unique to the Sequence Initiator and the respective Sequence Recipient pair (see 24.5). **(concurrent Exchanges and Sequences)**

h) The local F_Port exercises buffer-to-buffer flow control with the transmitting N_Port. The remote F_Port exercises buffer-to-buffer flow control with the receiving N_Port. R_RDY is used for buffer-to-buffer flow control (see 22.3.5). **(buff r to buff r flow control)**

i) If the Fabric is unable to deliver the frame to the destination N_Port, the frame is discarded and the source is not notified. If the destination N_Port is unable to receive the frame, the frame is discarded and the source is not notified. **(non-delivery)**

j) The buffer-to-buffer Credit is used for buffer to buffer flow control. End-to-end Credit is not used (see 26.3). Class 3 may share the Credit for Connectionless service with Class 2 and Class 1/**SOF**$_{c1}$ frames (see 26.3). **(Credit)**

k) Effective transfer rate between any given N_Port pair is dependent upon the number of N_Ports a given N_Port is demultiplexing to and multiplexing from. **(bandwidth)**

l) Neither the F_Port nor N_Port shall issue busy or reject to Class 3 frames. **(busy/reject)**

m) Frames within a Sequence are tracked on a Sequence_Qualifier and SEQ_CNT basis (see 18.1.1). **(tracking)**

n) An N_Port or F_Port shall be able to recognize **SOF** delimiters of all Classes of service, whether or not all Classes are supported by the Port, and provide appropriate responses for all Classes with appropriate delimiters. An N_Port which supports only Class 3 shall issue a P_RJT for Class 1 or Class 2 frames with appropriate Class 1 or Class 2 delimiters respectively while obeying the buffer-to-buffer flow control rules. An F_Port which supports only Class 3 shall issue a F_RJT for Class 1 or Class 2 frames with appropriate Class 1 or Class 2 delimiters respectively, while obeying the buffer-to-buffer flow control rules. **(invalid Class)**

o) An N_Port may obtain the delivery status of Class 3 Sequences transferred by using Abort Sequence protocol (see 29.7.1.1) and thus verify the integrity of the delivered Sequences. **(Sequence integrity)**

22.3.3 Class 3 delimiters

Sequences are initiated by transmitting a frame started by an **SOF**$_{i3}$. Subsequent frames within a Sequence are started by an **SOF**$_{n3}$. A Sequence is terminated with a Data frame ended by **EOF**$_{t}$. All frames other than the last frame within the Sequence are terminated by an **EOF**$_{n}$.

22.3.4 Class 3 fram size

The size of each frame transmitted is limited by the smaller value of the maximum Data Field size supported by the Fabric and by the receiving N_Port. Each frame is routed through the Fabric, if present, as a separate entity.

22.3.5 Class 3 flow control

Class 3 uses only buffer to buffer flow control with R_RDY. End-to-end flow control is not supported.

All Class 3 frames shall follow buffer-to-buffer flow control rules (see 26.5.1).

22.3.6 Class 3 Sequence integrity

With a missing Class 3 Data frame, the Sequence Recipient is capable of detecting the error of non-receipt of the frame, but can not communicate it to the Sequence Initiator due to absence of ACK in Class 3. However, using Abort Sequence protocol (see 24.3.11 and 29.7), the Sequence Initiator can verify if one or more transmitted Sequences were received without any Sequence error. This usage of Abort Sequence protocol makes it possible to verify the integrity of Class 3 Sequences delivered.

If a sending ULP relies on the receiving ULP for ensuring Sequence integrity, the Sequence Initiator may not use the Abort Sequence protocol to confirm Sequence delivery.

22.4 Intermix

Intermix is an option of Class 1 service which allows interleaving of Class 2 and Class 3 frames during an established Class 1 Connection between two N_Ports. While engaged in a Class 1 Connection, an N_Port capable of Intermix may transmit and receive Class 2 and Class 3 frames interleaved with Class 1 frames. Class 2 and Class 3 frames may be interchanged with either the N_Port at the other end of the Connection or with any other N_Port. An N_Port which supports Intermix shall be capable of both transmitting and receiving intermixed frames. Support for Intermix option of Class 1 service is specifi d during Login.

In a point-to-point topology, both interconnected N_Ports shall be required to support Intermix if Intermix is to be employed. In the presence of a Fabric, both the N_Port and the Fabric shall be required to support Intermix if Intermix is to be employed in that link. Fabric support for Intermix requires that the full Class 1 bandwidth during a Dedicated Connection be maintained. Intermix permits the potentially unused Class 1 bandwidth to be available for transmission of Class 2 and Class 3 frames.

22.4.1 Fabric management

If a Fabric that supports Intermix receives a Class 2 frame destined to an N_Port engaged in a Dedicated Connection and the N_Port does not support Intermix, the Fabric shall return an F_BSY to the source N_Port of the frame, after ensuring that the frame is not deliverable. The Fabric is allowed to hold a Class 2 frame for a period of time before issuing F_BSY. If the Fabric receives a Class 3 frame destined to an N_Port engaged in a Dedicated Connection and the N_Port does not support Intermix, the frame is discarded.

If a Fabric that does not support Intermix receives a Class 2 frame from a third N_Port destined to either of the N_Ports engaged in a Dedicated Connection, an F_BSY shall be returned to the source N_Port of the frame. If this Fabric receives a Class 3 frame from a third N_Port destined to either of the N_Ports engaged in a Dedicated Connection, the frame is discarded. The Fabric is allowed to hold a Class 2 frame for a period of time before issuing F_BSY.

If an N_Port, established in a Dedicated Connection and attached to a Fabric that does not support Intermix, transmits a Class 2 or Class 3 frame destined to the other N_Port of the Dedicated Connection or to a third N_Port, the destination of this frame delivery is unpredictable.

During an established Class 1 Connection with Intermix supported, Class 1 frames have priority over Class 2 or Class 3 frames. Class 2 and Class 3 frames shall be interleaved during a Class 1 Connection only if there is no backlog of Class 1 frames en route. Although a Fabric and an attached N_Port both support Intermix, the Fabric may choose to transmit F_BSY to a Class 2 frame or discard a Class 3 frame while th N_Port is engaged in a Class 1 Connection.

The Fabric shall provide adequate buffering for an incoming Class 1 frame while a Class 2 or Class 3 frame is being transmitted during a Class 1 Connection. The extent of buffering needed is dependent on the manner in which a Class 2 or Class 3 frame in transit is managed:

- to successfully complete a Class 2 or 3 frame in transit, an incoming Class 1 frame shall be buffered to the extent of the maximum Class 2 or Class 3 frame size, or
- if a Class 2 or Class 3 frame is being aborted (**EOFa**) on receipt of a Class 1 frame, the Class 1 frame shall be buffered to the extent of the time required to append an **EOFa** to the Class 2 or 3 frame in transit.

22.4.2 Intermix rules

An N_Port pair shall have a Class 1 Connection established for Intermix rules to be applicable to them. To provide an Intermix service, the Fabric and the receiving and transmitting N_Ports shall obey the following rules:

a) The Fabric or an N_Port shall provide the Intermix Service Parameter during Login to indicate its Intermix capability to the communicating Port. **(Intermix Service Parameter)**

b) If the Fabric supports Intermix, an N_Port supporting Intermix is allowed to transmit Class 2 and Class 3 frames during a Class 1 Connection intermixed with Class 1 frames of that Connection.

If the Fabric does not support Intermix, the N_Port shall not transmit intermixed frames. If the N_Port transmits intermixed frames, delivery of these frames is unpredictable. **(transmit Intermix)**

c) An attached N_Port supporting Intermix shall be capable of receiving Class 2 and Class 3 frames during Class 1 Connection intermixed with Class 1 frames of that Connection. If an N_Port supporting Intermix receives a Class 2 or Class 3 frame intermixed, and the Fabric does not support Intermix, the frame shall be discarded. **(receive Intermix)**

d) If the Fabric that supports Intermix, receives a Class 2 or Class 3 fram destined to an N_Port engaged in a Dedicated Connection, and the destination N_Port does not support Intermix, the Fabric shall return F_BSY to the Class 2 frame after ensuring that the frame is not deliverable. The Fabric shall discard the Class 3 frame without any response to the sender. **(Intermix Fabric busy or discard)**

e) If the Fabric that does not support Intermix, receives a Class 2 or Class 3 frame from a third N_Port destined to an N_Port engaged in a Dedicated Connection, regardless of whether the destination N_Port supports Intermix or not, the Fabric shall return F_BSY to the Class 2 frame after ensuring that the frame is not deliverable. The Fabric shall discard the Class 3 frame without any response to the sender.
If an N_Port attached to the Fabric that does not support Intermix transmits intermixed frames in violation of rule b, the delivery of these intermixed frames is unpredictable. **(Non-Intermix Fabric busy or discard)**

f) The Fabric shall guarantee full Class 1 bandwidth during a Dedicated Connection. Class 2 and Class 3 frames shall flow on the potentially unused Class 1 bandwidth. **(bandwidth sharing)**

g) Class 1 frames have priority over Class 2 and Class 3 frames. Class 2 and Class 3 frames may be intermixed only when there is no backlog of Class 1 frames. The Fabric may issue F_BSY to a Class 2 frame and discard a Class 3 frame. The Fabric may abort (**EOFa**) a Class 2 or Class 3 frame in progress if its transmission interferes with the transmission of a Class 1 frame. **(Class 1 precedence)**

h) The Fabric may cause a delay and displace a Class 1 frame in time due to Intermix. This delay is limited to the maximum Class 2 or 3 frame size. The Fabric shall provide adequate buffering for the incoming Class 1 frame while a Class 2 or a Class 3 frame is in transit and guarantee the integrity and delivery of Class 1 frames. **(Class 1 frame skew and integrity)**

i) In a point-to-point topology, an N_Port which supports Intermix may transmit and receive Class 2 and Class 3 frames during Class 1 Connection intermixed with Class 1 frames if the other N_Port supports Intermix. **(point-to-point Intermix)**

22.4.3 Intermix delimiters

Intermix does not impose any additional delimiters.

22.4.4 Intermix fram size

The size of each Class 1, Class 2, or Class 3 frame is governed by the limitations of each Class individually. Intermix does not impose any additional limitation on the frame size.

22.4.5 Intermix fl w control

The flow control for each Class is governed separately by individual Class. Intermix does not impose any additional rules on flow control.

23 Login and Service Paramet rs

23.1 Introduction

The Login procedure is a method by which an N_Port establishes its operating environment with a Fabric, if present, and other destination N_Ports with which it communicates. Fabric Login and destination N_Port Login are both accomplished with a similar procedure using different Destination_Identifiers (D_IDs) and possibly different Source_Identifiers (S_IDs).

Login between an N_Port and the Fabric or between two N_Ports is long-lived. The number of concurrent N_Ports with which an N_Port may be logged in with is a function of the N_Port facilities available. There is no one to one relationship between Login and Class 1 Dedicated Connections.

Explicit Login is accomplished using the Login (FLOGI/PLOGI) Link Service Sequence within a separate Exchange to transfer the Service Parameters (contained in the Payload) of the N_Port initiating the Login Exchange. The Accept (ACC) contains the Service Parameters of the Responder (contained in the Payload).

Implicit Login is a method of defining and specifying the Service Parameters of destination N_Ports by means other than the explicit use of the Login Exchange. Specific methods of **implicit** Login are not defined in FC-PH.

When Login is referred to throughout other sections of this document, either the **explicit** or **implicit** procedure is acceptable. Implicit Login is assumed to provide the same functionality as Explicit Login. Default Login values are specified in 21.1.1. Explicit Login replaces previous Service Parameters and initializes end-to-end or buffer-to-buffer Credit, or both. The Login protocol shall follow the Exchange and Sequence management rules as specified in clause 24. Frames within a Sequence shall operate according to R_RDY Primitive, ACK, and Link_Response rules specified in clause 20.

Explicit Fabric Login is performed during the initialization process under the assumption that a Fabric is present. The first explicit Login is directed to the Fabric using the well-known Fabric address (i.e., F_Port at hex 'FFFFFE'). It is mandatory for all Fabric types to support the explicit Login procedure. An N_Port shall use binary zeros in its first attempt at explicit Fabric Login as its S_ID. If it receives an F_RJT with a reason code of Invalid S_ID, it may use its last known native address identifier in the FLOGI Sequence as its S_ID.

Login with the Fabric provides the N_Port with Fabric characteristics for the entire Fabric as defined in the Fabric's Service Parameters. The Service Parameters specified by the N_Port provide the Fabric with information regarding the type of support the N_Port requests. The Service Parameters provided by an N_Port also include a 64-bit N_Port_Name and 64-bit Node_Name. The Service Parameters provided by an F_Port also include a 64-bit F_Port_Name and 64-bit Fabric_Name. During the Fabric Login procedure the Fabric may optionally define the N_Port's native Identifier within a system configuration. If the Fabric does not support native N_Port Identifier assignment, the N_Port shall assign its own native N_Port Identifier by another method not defined in FC-PH.

Destination N_Port Login (PLOGI) provides each N_Port with the other N_Port's Service Parameters. Knowledge of a destination N_Port's receive and transmit characteristics is required for data Exchanges. Service Parameters of destination N_Ports are saved and used when communication with those N_Ports is initiated. The Service Parameters interchanged between two N_Ports may be asymmetrical. Saving the Service Parameters of destination N_Ports with which an N_Port communicates requires N_Port resources. These resources can be released using the destination N_Port Logout procedure (see 23.5).

An N_Port may transmit the Link Credit Reset command to a destination N_Port in order to reclaim end-to-end Credit for outstanding Data frames in Class 2.

When an N_Port receives a Login from an N_Port, all other Active and Open Sequences with the N_Port performing reLogin are abnormally terminated prior to transmission of the Accept Sequence to the reLogin request.

23.2 Applicability

Login with the Fabric is required for all N_Ports, regardless of Class of Service supported. Communication with other N_Ports shall not be accepted until the Fabric Login procedure is complete. For an N_Port which supports Class 1 or Class 2 Service, an N_Port is required to Login with each N_Port with which it intends to communicate (destination N_Port Login).

NOTE - It is not required that an N_Port provide the same Login information with each destination N_Port or with the Fabric. However, an N_Port should avoid using contradictory or conflicting parameters with different Login destinations.

23.3 Fabric Login

Fabric Login accomplishes five functions

- It determines the presence or absence of a Fabric.
- If a Fabric is present, it provides a specific set of operating characteristics associated with the entire Fabric.
- If a Fabric is present, it shall optionally assign or shall confirm the native N_Port Identifier of the N_Port which initiated the Login.
- If a Fabric is not present, an Accept with the specification of N_Port in Common Service Parameters indicates that the requesting N_Port is attached in a point-to-point topology.
- If a Fabric is present, it initializes the buffer-to-buffer Credit.

23.3.1 Explicit Fabric Login

The explicit Fabric Login procedure shall require transmission of a Login (FLOGI) Link Service Sequence within an Exchange with an assigned OX_ID by an N_Port with a Destination Identifier (D_ID) of the well-known F_Port address (FFFFFE) and a Source Identifier (S_ID) of binary zeros in its first attempt at Fabric Login.

When S_ID = 0 is used, the F_Port shall either

- assign a Fabric unique N_Port Identifier to the N_Port in the ACC reply Sequence, or
- return an F_RJT with a reason code indicating Invalid S_ID, if the Fabric does not support N_Port Identifier assignment. The N_Port shall assign its native N_Port Identifier by another method not defined in FC-PH and r try the FLOGI Sequence with an S_ID = X.

If S_ID = X is used, the F_Port shall either

- return an ACC reply Sequence with D_ID = X (confirmed Identifier), or
- return an F_RJT with a reason code indicating Invalid S_ID, if X is invalid.

If the F_Port has rejected both S_ID = 0 and S_ID = X, the N_Port shall attempt to reLogin with another value of X, or determine a valid value of X by a method not defined in FC-PH.

The Payload of this Link Service Sequence contains the Service Parameters of the N_Port transmitting the FLOGI Sequence. The N_Port transmits Service Parameters as specified for F_Port Login which are defined in 23.6.2 and 23.6.7. Only those parameters associated with Fabric Login are specified (see table 101).

The normal reply Sequence to a FLOGI Link Service Sequence by an F_Port is an Accept (ACC) Link Service Sequence within an Exchange with the OX_ID of the Login request and F_Port assigned RX_ID with a Destination Identifier (D_ID) containing the N_Port Identifier assigned to the N_Port by the Fabric and a Source Identifier (S_ID) of the well-known F_Port address (FFFFFE). The Payload of the ACC contains the Service Parameters of the Fabric. Fabric Service Parameters are defined in 23.7.

The N_Port Common Service Parameters during Fabric Login are specified in 23.6.2. The N_Port Class Service Parameters during Fabric Login are specified in 23.6.7. The F_Port Common Service Parameters specified in the Accept during Fabric Login are specified in 23.7.1. The F_Port Class Service Parameters specified in the Accept during Fabric Login are specified in 23.7.4.

23.3.2 Responses to Fabric Login

The following meanings are associated with table 94 and table 95:

- the Response/Reply Seq column identifies R_RDY, a Link_Control response, or a Link Service frame transmitted in response to the FLOGI Sequence directed to the well-known F_Port address (FFFFFE). More than one frame is possible in response.

– the D_ID and S_ID columns specify the value of the corresponding field in the response frame received by the N_Port transmitting the FLOGI Sequence.

– the Indication column provides a short summary of the possible conditions associated with a particular response.

– the N_Port Action column specifies the action for the N_Port transmitting the FLOGI Sequence to take based on the response received.

23.3.2.1 FLOGI with S_ID = 0

Table 94 describes the set of possible responses to Fabric Login with an S_ID = 0 and a D_ID of hex 'FFFFFE'. Further description of the response primitive or frame is found in clause 20.

Table 94 - Responses to FLOGI frame (S_ID = 0) - Fabric Login

Response/ Reply Seq	D_ID	S_ID	Indication	N_Port Action
1.R_RDY	N/A	N/A	- Class 1(SOF_{c1}) - Class 2 or 3 - successful frame delivery to F_Port or N_Port	- wait for frame
2.ACK_1	0	FFFFFE	- FLOGI frame has been received by N_Port or F_Port	- Wait for Reply Data frame Sequence
3.ACC	X	FFFFFE	- OX_ID = FLOGI OX_ID -If Common Serv = F_Port, - Fabric Login complete	- Perform destination N_Port Login (23.4.2.1) (Fabric present)
4 ACC	0	FFFFFE	- OX_ID = FLOGI OX_ID -If Common Serv = N_Port - no Fabric present	- Perform point-to-point destination N_Port Login
5.F_BSY or P_BSY	0	FFFFFE	- Busy	- retry later
6.F_RJT or P_RJT	0	FFFFFE	- reason code = Class not supported = invalid S_ID = other	 - FLOGI -next Class (S_ID = 0) - FLOGI with S_ID = X - respond accordingly
7 FLOGI	FFFFFE	0	- Collision with other N_Port	- respond with ACC Common Serv = N_Port - compare N_Port_Names - if xmit P_Name > rec'd P_Name End Connection if Class 1 - initiate point-to-point destination N_Port Login - if xmit P_Name < rec'd P_Name End Connection if Class 1 - wait for point-to-point destination N_Port Login
8.LS_RJT	X or 0	FFFFFE	- Fabric present - reason code	 - reLogin with altered Service Parameters, use D_ID of LS_RJT
9.None			- error	- perform ABTS after Sequence timeout

These responses are characterized by the following:

- Response 1 is possible from an N_Port or Fabric.
- Response 2 is from a Fabric or an N_Port. The D_ID and S_ID values (in the response to the FLOGI frame) correspond to the values in the FLOGI fields, respectively, in the FLOGI frame (also for responses 5 and 6).
- Response 3 completes Fabric Login. The N_Port S_ID is assigned as X.
- Response 4 indicates a point-to-point topology with another N_Port: which is determined by examining the Common Service Parameter which specifies N_Port or F_Port. Based on comparison of Port_Names, either transmit PLOGI, or wait for PLOGI.
- Response 5 indicates that either the Fabric or N_Port is busy, retry later.
- Response 6 indicates a Fabric or N_Port reject. If Class is not supported, retry Login with another Class with a numerically higher value. If the reason code is S_ID invalid, then retry FLOGI with a value of X (see 23.3.2.2). For other reject reasons, the N_Port shall respond accordingly.
- Response 7 indicates a point-to-point attachment and a collision with an FLOGI from the attached N_Port. The N_Port shall respond with an ACC. The Common Service Parameters shall contain the same information as FLOGI but shall indicate that an N_Port is transmitting the Data. Port_Name and Node_Name shall be included, but all Classes of Service shall be indicated as invalid. The N_Port shall compare the Port_Name received to the Port_Name it transmitted. If this N_Port's value is lower, it shall end this Exchange and wait for a PLOGI from the attached N_Port. In Class 1 if this N_Port's value is lower, it shall become the Connection Recipient (see clause 28) for this Connection. In Class 1 if its value is higher, it shall become the Connection Initiator for this Connection. If its value is higher, it shall transmit a PLOGI (see 23.4.2.3) as part of a new Dedicated Connection. The Dedicated Connection associated with FLOGI shall be removed by the normal method to remove a Dedicated Connection in Class 1. See 23.6.4 for a description of N_Port_Names. See 23.4.2.3 for a description of point-to-point destination N_Port Login.
- Response 8 indicates that the Login request is being rejected for a reason specified in the LS_RJT frame. The FLOGI request may be retried if the appropriate corrective action is taken.
- Response 9 indicates that a Link error has occurred.

NOTE - When an N_Port originates an Exchange using an N_Port Identifier of unidentified (binary zeros), its N_Port Identifier may change between transmitting a request Sequence (Login) and receiving the reply Sequence (Accept).

23.3.2.2 FLOGI with SID = X

Table 95 describes the set of possible responses to Fabric Login with an S_ID = X. The FLOGI Sequence transmitted contains a D_ID of the well-known F_Port address (FFFFFE) and an S_ID of X. It is known that a Fabric is present before this Fabric Login is attempted.

Table 95 - Responses to FLOGI frame (S_ID = X) - Fabric Login				
Response/ Reply Seq	D_ID	S_ID	Indication	N_Port Action
1.R_RDY	N/A	N/A	- Class 1(SOFc1) - Class 2 or 3 - successful frame delivery to F_Port	- wait for frame
2.ACK_1	X	FFFFFE	- FLOGI frame has been received	- Wait for Reply Data frame Sequence
3.ACC	X	FFFFFE	- OX_ID = FLOGI OX_ID - Fabric Login complete - Address Identifier = X	- Perform destination N_Port Login
4.F_BSY	X	FFFFFE	- Fabric Busy	- retry later
5.F_RJT	X	FFFFFE	- reason code = invalid S_ID = other	- FLOGI with S_ID = different X - respond accordingly
6.LS_RJT	X	FFFFFE	- reason code	- FLOGI with altered Service Parameters
7.None			- error	- perform ABTS after Sequence timeout

These responses are characterized by the following·

— Response 1 is possible from a Fabric.

— Response 2 is from a Fabric. The D_ID and S_ID values (response to the FLOGI frame) correspond to the values in the FLOGI fields, respectively, in the FLOGI frame (also for responses 4 and 5).

— Response 3 completes Fabric Login. The N_Port S_ID is confirmed as X.

— Response 4 indicates that the Fabric is busy.

— Response 5 indicates a Fabric reject. If Class is not supported, retry Login with another Class, with a numerically higher value.

If the reason code is S_ID invalid, then retry FLOGI with a different value of X. For other reject reasons, the N_Port shall respond accordingly.

— Response 6 indicates that the Login request is being rejected for a reason specified in the LS_RJT frame. The FLOGI request may be retried if the appropriate corrective action is taken.

— Response 7 indicates that a Link error has occurred.

23.3.3 SOF delimiters

Since the Fabric may not support all three Classes of Service, the FLOGI Sequence may require retry of the FLOGI Sequence with a different **SOF** delimiter for each of the following Classes in a, b, and c order:

a) Class 1 - **SOFc1 (SOFi1)**
b) Class 2 - **SOFi2**
c) Class 3 - **SOFi3**

Selection of the **SOF** delimiter is based on the Classes of Service supported by the originating N_Port. The FLOGI Sequence is transmitted and the appropriate action is specified in table 94, or table 95. If a Reject (F_RJT, P_RJT) has been received indicating incorrect Class, the next consecutive, supported delimiter on the above list is attempted until the Login procedure is complete or all supported delimiter types have been attempted. When multiple service classes are desired, the first time that Login is successful, the Service Parameters for all applicable service classes have been processed. Login is only valid for the class used to Login and all Classes with higher numerical value (i.e., Class 1, 2, and 3).

NOTE - Class 1 communication requires that both N_Ports operate at the same speed. Hence, N_Port Login should be performed in Class 1 if that Class is supported. Class 2 or 3 communication does not require that both N_Ports operate at the same speed.

Link speed is not a Login parameter. A mismatch in Class 1 is indicated by an F_RJT with a reason code of "Fabric path not available".

If all supported delimiter types have been attempted and all have been rejected by the Fabric or timed out, the Fabric and N_Port are incompatible and outside intervention is required.

23.3.4 Frequency

Login between an N_Port and the Fabric should be long-lived. If Implicit Logout with the Fabric has occurred, it is necessary to reLogin with the Fabric (see 23.5.3).

23.3.4.1 Login completion - Originator

The Originator of the FLOGI request considers Login to have ended when

a) in Class 1, the Originator has transmitted the ACK (**EOFt** or **EOFdt**) to the Accept, or
b) in Class 2, the Originator has received the R_RDY in response to transmitting the ACK (**EOFt**) to the Accept, or
c) in Class 3, the Originator has transmitted the R_RDY in response to the Accept.

When Login is ended, the values of buffer-to-buffer and end-to-end Credit are initialized.

23.3.4.2 Login completion - Responder

The Responder of the FLOGI request considers Login to have ended when

a) in Class 1, the Responder has received the ACK (**EOFt** or **EOFdt**) to the Accept, or
b) in Class 2, the Responder has transmitted the R_RDY in response to receiving the ACK (**EOFt**) to the Accept, or
c) in Class 3, the Responder has received the R_RDY in response to the Accept.

When Fabric Login has ended successfully, the values of buffer-to-buffer and end-to-end Credit are initialized.

NOTE - Since the N_Port has not yet performed Fabric Login, it can not be doing anything else. Therefore R_RDY can be assumed to be part of the Fabric Login protocol.

23.4 N_P rt Login

Destination N_Port Login proceeds following the Fabric Login procedure. Events 4, 7, 8, and 9 in table 94 indicate a point-to-point attachment and the N_Port shall proceed based on the information provided by the N_Port action for each event. If a Fabric is present, as determined by performing the Fabric Login procedure, an N_Port proceeds with destination N_Port Login according to 23.4.2.1. If a Fabric is not present, as determined by performing the Fabric Login procedure, an N_Port proceeds with destination N_Port Login according to 23.4.2.3. Destination N_Port Login between two N_Ports is complete when each N_Port has received the Service Parameters of the other N_Port. This may be accomplished by either implicit or explicit destination N_Port Login.

The N_Port Common Service Parameters during N_Port Login are specified in 23.6.3. The N_Port Class Service Parameters during N_Port Login are specified in 23.6.8. Both the Common Service Parameters and Class Service Parameters apply to each N_Port during destination N_Port Login.

NOTE - When an N_Port (A) receives a PLOGI from another N_Port (B), N_Port (A) should verify that it is not already logged in with an N_Port (c) with the same Port_Name but different N_Port native address identifier and Node_Name. If so, it should consider the prior Login to be ended and should follow the Logout rules before accepting the new Login. Such a situation may arise if configuration changes have occurred.

23.4.1 Address Identifiers

An N_Port determines its own native N_Port Identifier through explicit or implicit Login by

- the Fabric, if present,
- implicit definition,
- assignment in the PLOGI Sequence transmitted to a destination N_Port attached in a point-to-point topology.

Address identifiers of other destination N_Ports with which an N_Port wishes to Login with may be collected from

- th Fabric, if present,
- a name-serv r function,
- implicit definition, or

– an alternate initialization procedure.

23.4.2 Explicit N_Port Login

Explicit N_Port Login accomplishes three functions:

– It provides a specific set of operating characteristics associated with the destination N_Port.

– Initializes the destination end-to-end Credit.

– In point-to-point topology, buffer-to-buffer Credit is initialized.

A well-behaved N_Port shall Logout with another N_Port prior to initiating a reLogin. However, if an N_Port receives or transmits a Login Link Service request Sequence with another N_Port, it shall respond to any other Exchanges with that N_Port as though a Logout had been previously performed. During the N_Port Login procedure other communication with the destination N_Port shall not be initiated or accepted. Once the N_Port Login procedure has been successfully completed, communication between the N_Ports may be initiated or accepted. That is, if N_Port (A) performs a PLOGI request with N_Port (B) and N_Port (B) transmits the ACC reply Sequence, then either N_Port (A) or N_Port (B) may initiate communication for other protocols. N_Port (B) shall not be required to transmit a PLOGI request Sequence to (A) unless it wishes to invalidate or alter the existing Login parameters.

23.4.2.1 Fabric present

The destination N_Port explicit Login procedure requires transmission of a Login (PLOGI) Link Service Sequence within an Exchange with an assigned OX_ID with a Destination Identifier (D_ID = Y) of the destination N_Port and a Source Identifier (S_ID = X) of originating N_Port. The Payload of this Sequence contains the Service Parameters of the N_Port originating the PLOGI Sequence. N_Port Service Parameters are defined in 23.6.

The normal reply Sequence to a PLOGI Link Service Sequence by an N_Port is an Accept (ACC) Link Service Sequence within the Exchange identified by the OX_ID of the Login Sequence and the RX_ID assigned by the Responder with a Destination Identifier (D_ID) of the originating N_Port (PLOGI Sequence) and a Source Identifier (S_ID) of the responding N_Port. The Payload of the ACC contains the Service Parameters of the responding N_Port.

If a collision occurs, such that N_Port (A) has transmitted a PLOGI to N_Port (B) and N_Port (A) receives a PLOGI from N_Port (B) before receiving the ACC from N_Port (B), N_Port (A) shall respond as though its PLOGI had never been transmitted. There shall be no special processing required.

Table 96 - Responses to PLOGI frame - N_Port Login (Fabric present)				
Response/ Reply Seq.	D_ID	S_ID	Indication	N_Port Action
1.R_RDY	N/A	N/A	- Class 1(SOF_{c1}) - Class 2 or 3 only - successful frame delivery to F_Port	- wait for frame
2.ACK_1	X	Y	- PLOGI frame has been received	- Wait for Reply Data frame Sequence
3.ACC	X	Y	- OX_ID = PLOGI OX_ID - Login complete	- End
4.F_BSY	X	Y	- Fabric present - Fabric Busy	- retry later
5.F_RJT	X	Y	- Fabric present - reason code = invalid D_ID = other	- determine Port_ID for Y - reattempt Login - respond accordingly
6.P_BSY	X	Y	- N_Port busy	- retry later
7.P_RJT	X	Y	- reason code = Class not supported = other	- select next Class - respond accordingly
8 LS_RJT	X	Y	- reason code = invalid parameters = other	- correct parameters - respond accordingly
9.None			- error	-perform ABTS after Sequence timeout

23.4.2.2 Responses to N_Port Login (Fabric)

See 23.3 2 for a description of the column meanings. The entries in table 96 are based on previous Login with a Fabric. Table 96 describes the set of possible responses during destination N_Port Login with a Fabric present. These responses are characterized by the following:

— Response 1 is possible from a Fabric.

— Response 2 is from the N_Port. The D_ID and S_ID values (in the N_Port response to the PLOGI frame) correspond to the values in the PLOGI fields, respectively, in the PLOGI frame (also for responses 6, 7, and 8).

— Response 3 completes N_Port Login.

— Response 4 indicates that the Fabric is busy.

— Response 5 indicates a Fabric reject. The N_Port shall respond according to the reject reason code specified.

— Response 6 indicates that the destination N_Port is busy.

— Response 7 indicates an N_Port reject. The N_Port shall respond according to the action and reason code of the P_RJT.

— Response 8 indicates that the Login request is being rejected for a reason specified in the LS_RJT frame. The PLOGI request may be retried if the appropriate corrective action is taken.

— Response 9 indicates that a Link error has occurred.

23.4.2.3 No Fabric present (point-to-point)

This procedure is based on the assumption that response 4 or 7 in table 94 was received during an attempted Fabric Login. The destination N_Port explicit Login procedure requires transmission of a Login (PLOGI) Link Service Sequence within an Exchange with an assigned OX_ID.

Since the detection of a point-to-point topology was determined by reception of an ACC or FLOGI with Common Service Parameters indicating an N_Port is directly attached,

— the N_Port waits for a PLOGI from the other N_Port if its N_Port_Name is less than the attached N_Port_Name, or

— the N_Port transmits a PLOGI with S_ID = X and D_ID = Y (X ≠ Y) and no collision with PLOGI is anticipated if its N_Port_Name is greater than the attached N_Port_Name.

The Payload of this frame contains the Service Parameters of the N_Port originating the PLOGI Sequence. N_Port Service Parameters are defined in 23.6. The Common Service Parameters are specified for point-to-point N_Port Login.

The normal reply Sequence to a PLOGI Link Service frame by an N_Port is an Accept (ACC) Link Service Sequence with the OX_ID of the Login Sequence and the assigned RX_ID of the Responder. The Payload of the ACC contains the Service Parameters of the responding N_Port.

Table 97 - Responses to PLOGI frame - N_Port Login (No Fabric, point to point)

Response/ Reply Seq.	D_ID	S_ID	Indication	N_Port Action
1.R_RDY	N/A	N/A	- Class 1(SOFc1) - Class 2 or 3 only - successful frame delivery to N_Port	- wait for frame
2.ACK_1	X	Y	- PLOGI frame has been received	- Wait for Reply Data frame Sequence
3. ACC	X	Y	- Login complete as X and Y - OX_ID = PLOGI OX_ID	- End
4.P_BSY	X	Y	- N_Port busy	- retry later
5.P_RJT	X	Y	- reason code = Class not supported = other	- select next Class - respond accordingly
6.LS_RJT	X	Y	- reason code = invalid parameters = other	- correct parameters - respond accordingly
7.PLOGI	R	Z	- Collision with other N_Port	- compare N_Port_Names - if xmit P_Name > rec'd P_Name respond with LS_RJT, - if xmit P_Name < rec'd P_Name process PLOGI from N_Port respond with ACC or LS_RJT
8.None			- error	- perform ABTS after Sequence timeout

23.4.2.4 Responses to N_Port Login (point-to-point)

See 23.3.2 for a description of the column meanings. Table 97 describes the set of possible responses to destination N_Port Login for a point-to-point configuration.

These responses are characterized by the following:

- Response 1 is possible from the N_Port.
- Response 2 is from the N_Port. The D_ID and S_ID values correspond to the values in the PLOGI Sequence (also for responses 4 and 5).
- Response 3 completes N_Port Login.
- Response 4 indicates that the destination N_Port is busy.
- Response 5 indicates an N_Port reject. The N_Port shall respond according to the action and reason code of the P_RJT.
- Response 6 indicates that the Login request is being rejected for a reason specified in the LS_RJT frame. The PLOGI request may be retried if the appropriate corrective action is taken.
- Response 7 indicates that the other N_Port which is attached in a point-to-point attachment has issued an N_Port Login. After comparing N_Port_Names, the N_Port shall issue an LS_RJT (Login already in progress) and transmit N_Port Login (if not previously transmitted), or process the N_Port PLOGI.
- Response 8 indicates that a Link error has occurred.

Following reception of an ACC to FLOGI indicating an N_Port attached in a point-to-point topology, only one N_Port should be transmitting PLOGI. The S_ID and D_ID values of PLOGI shall be non-zero values which shall be used until implicit or explicit Logout.

23.4.3 SOF delimiters

Since the destination N_Port may support any of three Classes of Service, the PLOGI Sequence may require retransmission with a different **SOF** for each Class in the same manner described for Fabric Login (see 23.3.3.). Login is only valid for the class used to Login and all Classes with a higher numerical value (i.e., Class 1, 2, and 3).

23.4.4 Frequency

The frequency of destination N_Port Login is installation dependent based on the frequency of configuration changes which may alter the N_Port Identifiers within an installation. Service Parameters of other N_Ports are retained until the next destination N_Port Login or until N_Port Logout is performed.

23.4.4.1 Login completion - Originator

The Originator of the PLOGI request considers Login to have ended when

a) in Class 1, the Originator has transmitted the ACK (**EOFt** or **EOFdt**) to the Accept, or
b) in Class 2, the Originator has received the Accept and transmitted the ACK (**EOFt**) to the Accept, or
c) in Class 3, the Originator has received the Accept.

When N_Port Login is ended with a Fabric present, the value of end-to-end Credit is initialized. When N_Port Login is ended in a point-to-point topology, the values of buffer-to-buffer and end-to-end Credit are initialized.

23.4.4.2 Login completion - Responder

The Responder of the PLOGI request considers Login to have ended when

a) in Class 1, the Responder has received the ACK (**EOFt** or **EOFdt**) to the Accept, or
b) in Class 2, the Responder has received the ACK (**EOFt**) to the Accept, or
c) in Class 3, the Responder has transmitted the Accept.

When N_Port Login is ended with a Fabric pres nt, the value of end-to-end Credit is initialized. When N_Port Login is ended in a point-to-point topology, the values of buffer-to-buffer and end-to-end Credit are initialized.

23.4.5 N_P rt Login frame flow

See annex P figures P.5, P.6, and P.7 for examples of frame flow for destination N_Port Login for Classes 1, 2, and 3.

23.5 Logout

23.5.1 Introduction

The destination Logout procedure provides a method for removing service between two N_Ports. Logout releases resources associated with maintaining Service with a destination N_Port. There is no explicit Logout procedure for the Fabric, however, implicit Logout may occur between an N_Port and the Fabric (see 23.5.3).

NOTE - Explicit Fabric Logout is not needed since the Fabric has no resources dedicated to an N_Port which could be usefully made available.

23.5.2 Explicit N_Port Logout

Logout is accomplished by transmitting a Logout (LOGO) Link Service request Sequence to a destination N_Port. The Logout procedure is complete when the responding N_Port transmits an ACC Link Service reply Sequence.

If an N_Port desires to explicitly Logout, the initiating N_Port shall terminate other Active Sequences that it initiated with the destination N_Port prior to performing Logout, otherwise, the state of other Active Sequences is unpredictable. If an N_Port receives a Logout request while another Sequence is Active which was initiated from the requesting N_Port, it may reject the Logout request using an LS_RJT (Link Service Reject).

After an explicit Logout is performed with an N_Port, the default Login Service Parameters specified in 21.1.1 shall be functional if Login was explicit. After an explicit Logout is performed with an N_Port, the implicit Login Service Parameters shall be functional if Login was implicit.

23.5.3 Implicit Logout

If an N_Port receives or transmits the NOS or OLS Primitive Sequence, it shall be implicitly logged out from the Fabric, if present, or attached N_Port in a point-to-point topology. Fabric Login shall be performed following implicit Logout. Communication with other N_Ports shall not be accepted until the Fabric Login procedure is complete.

During reLogin with the Fabric, if an N_Port's native identifier has changed since the last Fabric Login, the N_Port shall not initiate or accept communication with other N_Ports until it has explicitly logged out with each N_Port and performed N_Port Login.

During reLogin with the Fabric, if the N_Port detects that the F_Port_Name has changed since the last Fabric Login, the N_Port shall wait an R_A_TOV timeout period before initiating or accepting communication with other N_Ports (i.e., implicit N_Port Logout). The timeout period shall start when the N_Port detects the change in F_Port_Name. After waiting the timeout period, the N_Port shall Logout and reLogin other N_Ports to which it had been previously Logged-In.

If an N_Port receives OLS from the Fabric, the Fabric may be indicating configuration changes internal to the Fabric using the Online to Offline Protocol.

NOTE - If an N_Port is concerned that a partial Fabric Login may be in process using its link immediately preceding its attempted Fabric Login, it may wait an R_A_TOV in order to ensure that the response it receives from the F_Port during Fabric Login is associated with its Login request.

23.6 N_Port Service Parameters

The first 16 bytes of the Payload following the LS_Command Code shall specify Service Parameters common to all three Classes. The next eight bytes shall contain the N_Port_Name of the N_Port transmitting the Service Parameters. The next eight bytes shall contain the Node_Name of the Node which controls the N_Port. The next 48 bytes of the Payload following the Node_Name shall specify three sets of Class Service Parameters as shown in figure 59. The first 16-byte group shall specify Service Parameters for Class 1. The second 16-byte group shall specify the

Service Parameters for Class 2. The third 16-byte group shall specify the Service Parameters for Class 3. The fourth 16-byte group is reserved. The fifth 16-byte group specifies Vendor Version Level.

23.6.1 N_Port Common Service Parameters

N_Port Common Service Parameters represent Parameters which are common across all Classes supported by an N_Port.

Applicability

Table 98 identifies fields within the N_Port Common Service Parameters and specifies the applicability of those Parameters to N_Port Login and F_Port Login.

16 Bytes	8 Bytes	8 Bytes	16 Bytes	16 Bytes	16 Bytes	16 Bytes	16 Bytes
Common Service Parameters	Port Name	Node/ Fabric Name	Class 1 Service Parameters	Class 2 Service Parameters	Class 3 Service Parameters	Reserved	Vendor Version Level

Figure 59 - FLOGI, PLOGI, or ACC Payload

Table 98 - N_Port Common Service Parameter applicability

			N_Port Login Class			Fabric Login Class		
Service Parameter	Word	Bits	1	2	3	1	2	3
FC_PH Version	0							
Highest Version	0	31-24	y	y	y	y	y	y
Lowest Version	0	23-16	y	y	y	y	y	y
Buffer-to-Buffer Credit	0	15-0	y	y	y	y	y	y
Common features	1	31-16						
Continuously Increasing (C)	1	31	y	y	y	n	n	n
Random Relative Offset (O)	1	30	y	y	y	n	n	n
Reserved	1	29						
N_Port/F_Port (N)	1	28	y	y	y	y	y	y
Buffer-to-Buffer Receive Data Field Size	1	11-0	y	y	y	y	y	y
N_Port Total Concurrent Sequences	2	31-16	y	y	y	n	n	n
Relative Offset by Info Category	2	15-0	y	y	y	n	n	n
R_A_TOV	2	31-0	n	n	n	n	n	n
E_D_TOV	3	31-0	(PTP)	(PTP)	(PTP)	n	n	n

Notes
1 "y" indicates yes, applicable;
2 "n" indicates no, not applicable
3 PTP indicates applicable only for Point-to-point

23.6.2 N_Port Common Service Param ters - Fabric Login

N_Port Common Service Parameters used during Fabric Login are shown in figure 60.

Word	Bits 31	0
0	31 FC-PH Version HHHHHHHH LLLLLLLL	0 Buffer-to-Buffer Credit BBBBBBBB BBBBBBBB
1	31 Common Features rrrNrrrr rrrrrrrr	0 Buffer-to-Buffer Rcv Data Field Size rrrrFFFF FFFFFFFF
2	31 Reserved rrrrrrrr rrrrrrrr	0 Reserved rrrrrrrr rrrrrrrr
3	31 Reserved rrrrrrrr rrrrrrrr	0 Reserved rrrrrrrr rrrrrrrr

Figure 60 - N_Port Common Service Parameters (Fabric Login)

23.6.2.1 FC-PH version

The FC-PH version field provides a method of determining the version of FC-PH which a Port shall be capable of supporting. The FC-PH field is divided into two one-byte fields. Table 99 indicates the hexadecimal values for the low and high FC-PH version levels in word 0 bits 31-16.

Table 99 - FC-PH Version - N_Port

Hex value	Version level
00	None
01 - 05	Reserved
06	4.0
07	4.1
08	4.2
09	4.3
others	Reserved

A version of FC-PH shall be assigned a new binary value when its implementation is not compatible with the previous version specified. An N_Port shall support all versions between the lowest and the highest levels specified.

- **Word 0, Bits 31 - 24 - Highest version (H)**

The binary value of bits 31-24 are encoded to represent the highest or most recent version of FC-PH which the Port shall be capable of supporting.

- **Word 0, Bits 23 - 16 - Lowest version (L)**

The binary value of bits 23-16 are encoded to represent the lowest or earliest version of FC-PH that the Port shall be capable of supporting.

Where there is overlap in the FC-PH versions supported, each Port shall operate in the highest FC-PH version mutually supported.

23.6.2.2 Buffer-to-buffer Credit

The buffer-to-buffer Credit (B) field (Word 0, bits 15-0) specified shall be associated with the number of buffers available for holding Class 1 connect-request, Class 2, or Class 3 frames received from the F_Port.

The buffer-to-buffer Credit (B) shall be a single value which represents the total buffer-to-buffer Credit available for all Classes. An N_Port tracks buffer-to-buffer Credit as a single entity for all frames subject to buffer-to-buffer flow control (see clause 26).

Values in the buffer-to-buffer Credit field have the same format as in Class end-to-end Credit Service Parameters.

23.6.2.3 Common features

- **Word 1, Bit 28 - N_Port/F_Port (N)**

  0 = N_Port
  1 = F_Port

Word 1 bit 28 shall be set to zero.

23.6.2.4 Buffer-to-buffer Data_Field size

- **Word 1, Bits 15-0 Buffer-to-buffer Receive Size (F)**

The **buffer-to-buffer Receive Data_Field Size** is a binary value (bits 15-0) which specifies the largest Data_Field Size for an FT_1 frame (17.4) that can be received by th N_Port supplying the

Service Parameters as a Sequence Recipient for:

- a connect-request (**SOFc1**),
- a Class 2 Data frame, or
- a Class 3 Data frame.

Values less than 128 or greater than 2 112 are invalid. Values shall be a multiple of four bytes. An N_Port shall support a Data Field size of at least 128 bytes, however, a minimum of 256 bytes is recommended.

The buffer-to-buffer Receive Data_Field size is specified by FC-2.

23.6.3 N_Port Common Service Parameters - N_Port Login

N_Port Common Service Parameters used during N_Port Login are shown in figure 61.

Word	Bits 31	0
0	FC-PH Version HHHHHHHH LLLLLLLL	Buffer-to-Buffer Credit (Pt to Pt) BBBBBBBB BBBBBBBB
1	Common Features COVNrrrr rrrrrrrr	Buffer-to-Buffer Rcv Data Field Size rrrrFFFF FFFFFFFF
2	Total Concurrent Sequences rrrrrrrr TTTTTTTT	Relative Offset by Info Category DDDDDDDD DDDDDDDD
3	E_D_TOV (Pt-to-Pt) tttttttt tttttttt	tttttttt tttttttt

Figure 61 - N_Port Common Service Parameters (N_Port Login)

23.6.3.1 FC-PH version

The FC-PH version field provides a method of determining the version of FC-PH which an N_Port shall be capable of supporting. The FC-PH field is divided into two one-byte fields. Table 100 indicates the hexadecimal values for the low and high FC-PH version levels in word 0 bits 31-24, and bits 23-16.

Table 100 - FC-PH Version lev l - N_Port

Hex value	Version level
00	None
01 - 05	Reserved
06	4.0
07	4.1
08	4.2
09	4.3
others	Reserved

A version of FC-PH shall be assigned a new binary value when its implementation is not compatible with the previous version specified. An N_Port shall support all versions between the lowest and the highest levels specified.

- **Word 0, Bits 31 - 24 - Highest version supported (H)**

The binary value of bits 31-24 are encoded to represent the highest or most recent version of FC-PH which the N_Port shall be capable of supporting.

- **Word 0, Bits 23 - 16 - Lowest version supported (L)**

The binary value of bits 23-16 are encoded to represent the lowest or earliest version of FC-PH that the N_Port shall be capable of supporting.

Where there is overlap in the FC-PH versions supported, each N_Port shall operate in the highest FC-PH version mutually supported.

23.6.3.2 Buffer-to-buffer Credit

Word 0, bits 15-0 shall only be meaningful for an N_Port in a point-to-point topology during Login between two N_Ports. In a topology other than point-to-point, word 0, bits 15-0 shall not be meaningful. The buffer-to-buffer Credit (B) field specified shall be associated with the number of buffers available for holding Class 1 connect-request, Class 2, or Class 3 frames received from the N_Port.

The buffer-to-buffer Credit (B) shall be a single value which represents the total buffer-to-buffer Credit available for all Classes. An N_Port tracks buffer-to-buffer Credit as a single entity for all frames subject to buffer-to-buffer flow control (see clause 26).

Values in the buffer-to-buffer Credit field have the same format as in the end-to-end Credit Class Service Parameters.

23.6.3.3 Common features

- **Word 1, Bit 31 - Continuously Increasing Offset (C)**

  0 = not supported
  1 = supported

Word 1, bit 31 = 1 indicates that the N_Port supplying this parameter shall be capable of supporting Continuously Increasing Relative Offset, if present (F_CTL bit 3), within a Sequence on a frame by frame SEQ_CNT basis. Bit 31 shall only be applicable to those Information Categories in which an N_Port supports Relative Offset (i.e., word 2, bits 15-0). See 3.1 and clause 27 for an explanation of Continuously Increasing Relative Offset.

Word 1, bit 31 support shall be applicable as a Sequence Initiator as well as a Sequence Recipient for all Classes of Service supported by the N_Port. However, support as a Sequence Initiator shall not require that transmitted Sequences use Relative Offset, even if both N_Ports provide such support.

- **Word 1, Bit 30 - Random Relative Offset (O)**

  0 = not supported
  1 = supported

Word 1, bit 30 = 1 indicates that the N_Port supplying this parameter shall be capable of supporting Random Relative Offset values, if present (F_CTL bit 3). Random values may increase, decrease, or otherwise fluctuate within a Sequence. Bit 30 shall only be applicable to those Information Categories in which an N_Port supports Relative Offset (i.e., word 2, bits 15-0). See 3.1 and clause 27 for an explanation of Random Relative Offset.

Word 1, bit 30 support shall be applicable as a Sequence Initiator as well as a Sequence Recipient for all Classes of Service supported by the N_Port. However, support as a Sequence Initiator shall not require that transmitted Sequences use Relative Offset, even if both N_Ports provide such support.

- **Word 1, Bit 29 - Valid Vendor Version Level (V)**

  0 = not valid
  1 = valid

Word 1, bit 29 = 1 indicates that the Vendor Version Level (fifth 16-byte group) contains valid information. If Word 1, bit 29 = 0, it indicates that the Vendor Version Level field is not meaningful.

- **Word 1, Bit 28 - N_Port/F_Port (N)**

  0 = N_Port
  1 = F_Port

Word 1 bit 28 shall be set = 0.

23.6.3.4 Buffer-to-buffer Data_Field size

- **Word 1, Bits 15-0 Buffer-to-buffer Receive Size (F)**

The **buffer-to-buffer Receive Data_Field Size** is a binary value (bits 15-0) which specifies the largest Data_Field Size for an FT_1 frame (17.4) that can be received by the N_Port supplying the Service Parameters as a Sequence Recipient for:

- a connect-request (**SOFc1**),
- a Class 2 Data frame, or
- a Class 3 Data frame.

Values less than 128 or greater than 2 112 are invalid. Values shall be a multiple of four bytes. An N_Port shall support a Data Field size of at least 128 bytes, however, a minimum of 256 bytes is recommended.

The buffer-to-buffer Receive Data_Field size is specified by FC-2.

23.6.3.5 Total Concurrent Sequences

Total Concurrent Sequences is the total number of Concurrent Sequences for all 3 classes that an N_Port is capable of supporting as a Recipient.

- **Word 2, Bits 23 - 16 - Total C ncurrent Sequences (T)**

The Total Concurrent Sequences specified by an N_Port shall be less than or equal to the sum of the Concurrent Sequences supported on a Class by Class basis. As an example, an N_Port may specify that it is capable of supporting ten Concurrent Sequences in Class 2 and ten Concurrent Sequences in Class 3. However, the total number of Concurrent Sequences when both Class 2 and 3 are Active may be fifteen (T).

23.6.3.6 Relative Offset by category

Word 2, bits 15 - 0 shall indicate on a bit-position basis, whether or not Relative Offset shall be supported for the corresponding Information Category. For example, if bit 14 = 1 and bit 2 = 1 and others are = 0, then Information Category 1110 and Information Category 0010 frames shall be capable of using Relative Offset as a Sequence Recipient or a Sequence Initiator.

23.6.3.7 Point-to-point E_D_TOV value

Word 3 shall only be meaningful by an N_Port in a point-to-point topology. In a topology other than point-to-point, word 3 shall not be meaningful. The E_D_TOV value shall be specified as a count of 1 ms increments. Therefore, a value of hex '0000000A' specifies a time period of 10 ms.

The E_D_TOV value in the Accept shall be greater than or equal to the value in the PLOGI. The E_D_TOV value in the Accept shall be the value used by each N_Port. R_A_TOV shall be a value twice the E_D_TOV value in a point-to-point topology.

23.6.4 N_Port_Name

The N_Port_Name is an eight byte field which identifies an N_Port for identification purposes, such as diagnostics, which may be independent and unrelated to network addressing. The format of the name is specified in 19.3. Each N_Port shall provide a unique N_Port_Name within the address domain of the Fabric. Bits 63-60 specify the format of the name. The formats are summarized in tables 43 and 44 in 19.3.

23.6.5 Node_Name

The Node_Name is an eight byte field which identifies a Node for identification purposes, such as diagnostics, which is independent and unrelated to network addressing. The format of the Node_Name is specified in 19.3. Each Node shall provide a unique Node_Name within the address domain of the Fabric. Bits 63-60 specify the format of the name. The formats are summarized in tables 43 and 44 in 19.3.

23.6.6 N_Port Class Service Parameters

Each group of sixteen byte Class Service Parameters is divided into the categories as specified in figure 62.

- Class Validity (V)
- Service Options (E)
- Initiator Control Flags (D)
- Recipient Control Flags (C)
- Receive Data Size (N)
- Concurrent Sequences (L)
- N_Port End-to-end Credit (M)
- Open Sequences per Exchange (X)

Word / Bits	31	0
0	Service Options VEEEEEEE EEEEEEEE	Initiator Control DDDDDDDD DDDDDDDD
1	Recipient Control CCCCCCCC CCCCCCCC	Receive Data Field Size rrrrNNNN NNNNNNNN
2	Concurrent Sequences rrrrrrrr LLLLLLLL	N_Port End-to-end Credit 0MMMMMMM MMMMMMMM
3	Open Sequences per Exchange rrrrrrrr XXXXXXXX	Reserved rrrrrrrr rrrrrrrr

Figure 62 - N_Port Class Service Parameters

The parameters described are located in the Payload of the Login (LOGI) Link Service request as well as the Accept (ACC) Link Service reply Sequence sent in reply to the Login Link Service request.

Applicability
Table 101 identifies N_Port Class Service Parameter fields and specifies th applicability of those parameters for N_Port Login or Fabric Login by class.

Table 101 - N_Port Class Service Parameter Applicability								
			N_Port Login Class			Fabric Login Class		
Service Parameter	Word	Bits	1	2	3	1	2	3
Class Validity	0							
Valid = 1	0	31	y	y	y	y	y	y
Service Options	0	30-16						
Intermix Mode	0	30	y	n	n	y	n	n
Stacked Connect-Requests	0	29-28	n	n	n	y	n	n
Sequential delivery	0	27	n	n	n	n	y	y
Initiator Ctl	0	15-0						
X_ID reassignment	0	15-14	y	y	n	n	n	n
Initial Responder Process_Associator	0	13-12	y	y	y	n	n	n
ACK_0 capable	0	11	y	y	n	n	n	n
ACK_N capable	0	10	y	y	n	n	n	n
Recipient Ctl	1	31-16						
ACK_0 Capable	1	31	y	y	n	n	n	n
ACK_N Capable	1	30	y	y	n	n	n	n
X_ID interlock	1	29	y	y	n	n	n	n
Error policy support	1	28-27	y	y	y	n	n	n
Categories per Sequence	1	25-24	y	y	y	n	n	n
Reserved - Fabric Specific	1	19-16	y	y	y	y	y	y
Receive Data Field Size	1	15-0	y	n	n	n	n	n
Concurrent Sequences	2	31-16	y	y	y	n	n	n
N_Port End-to-end Credit	2	14-0	y	y	n	n	n	n
Open Sequences per Exchange	3	31-16	y	y	y	n	n	n

Notes
1 "y" indicates yes, applicable;
2 "n" indicates no, not applicable

23.6.7 N_Port Class Service Parameters - Fabric Login

23.6.7.1 Class validity (V)

- **Word 0, Bit 31 - Class validity**

  0 = Invalid (Class not supported)
  1 = Valid (Class supported)

The Class validity bit shall indicate whether this Class is supported. If the Class validity bit is zero, it indicates that this group or set of sixteen bytes shall be ignored. If the Class validity bit is one, it indicates that this Class shall be supported. The Class shall be identified based on Class 1 = first sixteen-byte group, Class 2 = second sixteen-byte group, and Class 3 = third sixteen-byte group.

NOTE - Service Parameter options specify FC-2 capability. The Link Service may further limit values supplied during Login as specified by individual Upper Level Protocols.

23.6.7.2 Service options

Service options (E) shall specify optional features of a Class of Service supported by the N_Port supplying the Service Parameters.

Word,Bits	Service Options (E)
0,30-16	Class Options
0,30	Intermix Mode
0,29-28	Stacked Connect-Requests
0,27	Sequential delivery
0,26-16	Reserved

Figure 63 - Service options

- **Word 0, Bit 30 - Intermix Mode**

  0 = Intermix not requested
  1 = Intermix requested

Class 1

All N_Ports supporting Class 1 shall support Exclusive Connections. An N_Port supporting Exclusive Connections may only transmit and receive frames from the N_Port to which an existing Class 1 Connection is pending or established. Exclusive Connections require that the Fabric transmit an F_BSY frame in response to Class 2 frames and connect-request Data frames (SOF_{c1}) issued by a third N_Port targeted for one of the two N_Ports engaged in a Class 1 Connection.

An Intermixed Dedicated Connection specifies that the Fabric may insert or extract Class 2 or Class 3 frames while a Class 1 Connection is established. Support for Intermix is optional by both an N_Port and a Fabric. When an N_Port performs Login with a Fabric, it shall request support for Intermix by specifying bit 30 = 1. If the Fabric supplies, bit 30 = 1 in the Accept reply Sequence, then Intermix shall be functional.

The following set of values specifies the meaning of the combination of Word 0, bit 30:

N_Port	F_Port	
0	0	Neither supports
0	1	Fabric is capable of supporting, Intermix not functional
1	0	N_Port support requested, Fabric does not support
1	1	N_Port requested, Fabric agrees, Intermix is functional

See 22.4 for more discussion on Intermix.

Class 2 and 3

Word 0, bit 30 is reserved.

- **Word 0, Bits 29-28 - Stacked connect-requests**

Support for stacked connect-requests is optional in both a Fabric and an N_Port. Both an N_Port's and F_Port's behavior change if stacked connect-requests are functional.

Stacked connect-requests require Intermix mode support. Therefore, an N_Port shall only request support for stacked connect-requests if it also requests support for Intermix mode. Support for stacked connect-requests allows an N_Port to transmit one or more connect-requests (**SOFc1**) without regard as to whether a Connection is pending, established, or complete.

Due to the timing relationships involved, there are two methods of implementing stacked connect-requests by a Fabric:

a) transparent mode, or
b) lock-down mode.

In transparent mode, when the **SOFc1** Data frame is delivered to the destination N_Port, the return path of the bi-directional circuit is established in the same manner as Exclusive Dedicated Connections. This means that the destination N_Port of the **SOFc1** is able to transmit Data frames immediately following transmission of the ACK frame in response to the **SOFc1** frame.

In lock-down mode, when the **SOFc1** Data frame is delivered to the destination N_Port, the return path of the bi-directional circuit is not necessarily established to the source N_Port of the **SOFc1**. Therefore, the **SOFc1** Data frame shall also set F_CTL bit 8 = 1 (Unidirectional Transmit) in order to inhibit the destination N_Port of the **SOFc1** from sending any Data frames after the ACK frame is transmitted in response to the connect-request (see 28.5.2 for more information on stacked connect-requests and 18.5 for more information on F_CTL bit 8).

Bit 29 is a request by the N_Port for transparent stacked connect-requests. Bit 28 is a request by the N_Port for lock-down stacked connect-requests. An N_Port may request either one or both modes. However, a Fabric shall support either transparent or lock-down or none. The Fabric shall not support both modes.

Word 0, bit 29 Transparent mode - stacked connect-request

0 = Transparent mode not requested
1 = Transparent mode requested

Class 1

When an N_Port performs Login with a Fabric, it shall request support for transparent stacked connect-requests by specifying bit 29 = 1. Bit 29 = 1 is only meaningful if Intermix has also been requested. If the Accept reply from the Fabric specifies bit 29 = 1, then both the N_Port and Fabric have agreed that transparent stacked connect-requests are functional.

The following set of values specifies the meaning of the combination of Word 0, bit 29:

N_Port	F_Port	
0	0	Neither supports
0	1	Fabric is capable of supporting, transparent mode stacked connect-requests not functional
1	0	N_Port support requested, Fabric does not support
1	1	N_Port requested, Fabric agrees, transparent mode stacked connect-requests are functional

Class 2 and 3

Word 0, bit 29 is reserved.

Word 0, bit 28 Lock-down mode - stacked connect-request

0 = Lock-down mode not requested
1 = Lock-down mode requested

Class 1

When an N_Port performs Login with a Fabric, it shall request support for lock-down stacked connect-requests by specifying bit 28 = 1. Bit 28 = 1 is only meaningful if Intermix has also been requested. If the Accept reply from the Fabric specifies bit 28 = 1, then both the N_Port and Fabric have agreed that lock-down stacked connect-requests are functional.

The following set of values specifies the meaning of the combination of Word 0, bit 28:

```
N_Port  F_Port
  0       0    Neither supports
  0       1    Fabric is capable of supporting,
               lock-down mode stacked
               connect-requests not functional
  1       0    N_Port support requested,
               Fabric does not support
  1       1    N_Port requested, Fabric agrees,
               lock-down mode stacked
               connect-requests are functional
```

Class 2 and 3

Word 0, bit 28 is reserved.

- **Word 0, Bit 27 - Sequential delivery**

  0 = Out of order delivery
  1 = Sequential delivery requested

This bit is only meaningful in class 2 and 3. Out of order frame delivery in class 2 and 3 is the default function by a Fabric.

If bit 27 is set to one by an N_Port, then it is requesting that all frames delivered to the N_Port requesting this function be delivered in the same order in which the frames were transmitted by the source N_Port. If bit 27 is set to one by the F_Port in the Accept reply Sequence, the F_Port is indicating that sequential delivery is functional.

A Fabric supporting the sequential delivery feature routes Class 2 and 3 frames via a fixed route through the Fabric to provide in-order frame delivery. This feature does not imply any other alteration to the normal class of service functions. For example, F_BSY responses are still possible in Class 2, and Class 3 frames may still be discarded by the Fabric based on normal Class 2 and 3 rules.

Class 1

Bit 27 has no meaning in Class 1 since the Fabric shall deliver frames in the order transmitted based on class of service.

Class 2

In Class 2, if bit 27 = 1, the Fabric guarantees the order of delivery of both Data and Link_Control frames to the N_Port requesting this feature in the same order in which the frames were transmitted.

Class 3

In Class 3, if bit 27 = 1, the Fabric guarantees the order of delivery of Data frames to the N_Port requesting this feature in the same order in which the frames were transmitted.

```
N_Port  F_Port
  0       0    Neither supports
  0       1    Fabric is capable of supporting,
               Sequential delivery is functional
  1       0    N_Port support requested,
               Fabric does not support
  1       1    N_Port requested, Fabric agrees,
               Sequential delivery
               is functional
```

23.6.7.3 Initiator control

This field is not meaningful during Fabric Login.

23.6.7.4 Recipient control

This field is not meaningful during Fabric Login.

23.6.7.5 Receive Data_Field Size

This field is not meaningful during Fabric Login.

23.6.7.6 Concurrent Sequences

This field is not meaningful during Fabric Login.

23.6.7.7 N_Port End-to-end Credit

This field is not meaningful during Fabric Login.

23.6.7.8 Open Sequences per Exchange

This field is not meaningful during Fabric Login.

23.6.8 N_Port Class S rvic Parameters - N_Port Login

23.6.8.1 Class validity (V)

• **Word 0, Bit 31 - Class validity**

0 = Invalid (Class not supported)
1 = Valid (Class supported)

The Class validity bit shall indicate whether this Class is supported. If the Class validity bit is zero, it indicates that this set of sixteen bytes shall be ignored. If the Class validity bit is one, it shall indicate that this Class is supported. The Class shall be identified based on Class 1 = first sixteen-byte group, Class 2 = second sixteen-byte group, and Class 3 = third sixteen-byte group.

NOTE - Service Parameter options specify FC-2 capability. The Link Service may further limit values supplied during Login as specified by individual Upper Level Protocols.

23.6.8.2 Service options

Service Options (E) shall specify optional features of a Class of Service supported by the N_Port supplying the Service Parameters.

Word,Bits	Service Options (E)
0,30-16	Class Options
0,30	Intermix Mode
0,29-28	Stacked Connect-Requests
0,27	Sequential delivery
0,26-16	Reserved

Figure 64 - Service options

• **Word 0, Bit 30 - Intermix mod**

0 = Intermix not functional
1 = Intermix is functional

Class 1

When Word 0 bit 30 is set = 1 in N_Port Login with another N_Port, it indicates that Intermix is functional between the N_Port supplying this parameter and the Port to which it is attached. In a point-to-point topology if both N_Ports indicate Intermix support, then Intermix is functional. Otherwise, Class 1 Dedicated Connections shall be removed before transmission of Class 2 or Class 3 frames.

Class 2 and 3

Word 0, Bit 30 is reserved.

• **Word 0, Bits 29-28 - Stacked connect-requests**

Word 0, bits 29-28 are not meaningful in N_Port Login.

• **Word 0, Bit 27 - Sequential delivery**

This bit is not meaningful in N_Port Login.

23.6.8.3 Initiator control

Initiator Control Flags (D) specify which protocols, policies, or functions the Sequence Initiator function in the N_Port supplying the Service Parameters requests of the Recipient or is capable of as a Sequence Initiator.

Word,Bits	Initiator Ctl Flags (D)
0, 15-14	X_ID reassignment required
0, 13-12	Initial Responder Process_Associator
0, 11	ACK_0 support
0, 10	ACK_N support

Figure 65 - Initiator Ctl flags (D)

- **Word 0, Bits 15-14 - X_ID reassignment**

  0 0 = X_ID reassignment not supported
  0 1 = X_ID reassignment supported
  1 0 = Reserved
  1 1 = X_ID reassignment required (and supported)

X_ID reassignment required indicates that the N_Port supplying this parameter may reassign its X_ID value (either OX_ID or RX_ID) at certain Sequence boundaries (see 25.3.2) in Class 1 and 2. If X_ID reassignment is required, an N_Port shall transmit an Association Header at the beginning of the Exchange and at X_ID reassignment points within the Exchange. An N_Port which only supports X_ID reassignment shall use the Association Header as described in 25.3.1 when communicating with an N_Port which does require X_ID reassignment. An N_Port which requires X_ID reassignment shall also be able to support an N_Port which requires X_ID reassignment.

If the Responder N_Port to the PLOGI request requires X_ID reassignment and the Originator of the PLOGI request does not support X_ID reassignment, the Responder shall transmit an LS_RJT indicating the Initiator Control bits are in conflict. If the Responder N_Port to the PLOGI request does not support X_ID reassignment and the Originator of the PLOGI request has indicated that X_ID reasssignment is required, the Responder shall transmit an LS_RJT indicating the Initiator Control bits are in conflict. In either of these cases, the N_Ports are unable to communicate. If X_ID reassignment is required or supported, then X_ID interlock is also required.

- **Word 0, Bits 13-12 - Initial Process_Associator**

  0 0 = Initial Process_Associator not supported
  0 1 = Initial Process_Associator supported
  1 0 = Reserved
  1 1 = Initial Process_Associator required (and supported)

Initial Processs_Associator required indicates that the N_Port supplying this parameter requires an Association Header at certain Sequence boundaries (se 19.4) which contains a specific initial value in the Process_Associator field. An N_Port which supports Initial Process_Associator shall supply an Association Header with an initial Responder Process_Associator value at certain Sequence boundaries, such as when it originates an Exchange (see 25.3.1). If an Initial Process_Associator is required or supported, then X_ID interlock also is required in Class 1 and 2.

If the Responder N_Port to the PLOGI request requires an Initial Process_Associator and the Originator of the PLOGI request does not support an Initial Process_Associator, the Responder shall transmit an LS_RJT indicating the Initiator Control bits are in conflict. If the Responder N_Port to the PLOGI request does not support an Initial Process_Associator and the Originator of the PLOGI request has indicated that Initial Process_Associator is required, the Responder shall transmit an LS_RJT indicating the Initiator Control bits are in conflict. In either of these cases, the N_Ports are unable to communicate.

- **Word 0, Bit 11 - ACK_0 capability**

  0 = ACK_0 incapable
  1 = ACK_0 capable

Bit 11 specifies that the N_Port supplying these Class Service Parameters is or is not capable of support for ACK_0 as a Sequence Initiator for acknowledgement of an entire Sequence in either Discard or Process Exchange Error Policies. As a Sequence Initiator an N_Port receives and processes ACK frames in response to Data frame transmission. ACK_0 support is applicable to Class 1 and 2, but not Class 3. ACK_1 support is mandatory, whereas support for the ACK_0 form of ACK is optional.

N_Port A	N_Port B	
Word 0 Bit 11	Word 1 Bit 31	N_Port A as Sequence Initiator:
0	0	ACK_0 not supported
0	1	ACK_0 not supported
1	0	ACK_0 not supported
1	1	ACK_0 supported

If one N_Port (N_Port A, for example) is capable of receiving ACK_0 as a Sequence Initiator (Word 0, Bit 11 = 1) and the other N_Port (N_Port B, for example) is capabl of transmitting ACK_0 as a Sequence Recipient (Word 1, Bit 31 = 1), then ACK_0 is supported when N_Port A is the Sequence Initiator and N_Port B is the Sequence Recipient. Otherwise, ACK_0 shall not

be supported while N_Port A is the Sequence Initiator and N_Port B is the Sequence Recipient. ACK_0 usage shall take precedence over ACK_N and ACK_1.

If ACK_0 is supported, as indicated, N_Port A shall transmit Data frames to N_Port B with infinite buffering in effect and shall be able to receive ACK_0 for Sequences transmitted. If ACK_0 is supported, as indicated, N_Port B shall support infinite buffering and be capable of transmitting ACK_0 to acknowledge an entire Sequence (see 20.3.2.2 and 26.4.3.1).

ACK_0 capability may be asymmetrical for a single N_Port. That is, an N_Port may be capable processing ACK_0 as a Sequence Initiator, but not be capable of ACK_0 transmission as a Sequence Recipient. Similarly, an N_Port may be capable of generating ACK_0 as a Sequence Recipient, but not be capable of ACK_0 reception as a Sequence Initiator.

ACK_0 may be used for all forms of Discard Error Policies for Exchanges. ACK_0 support shall be required for Process Policy with infinite buffers Exchange Error Policy. However, any requirements regarding symmetrical or asymmetrical ACK_0 support shall be specified by the FC-4 or upper level and is outside the scope of FC-PH. For example, if an FC-4 only transmitted Video_Data from N_Port A to N_Port B (a frame buffer), then N_Port A need only support ACK_0 as a Sequence Initiator, whereas, N_Port B need only support ACK_0 as a Sequence Recipient. See 23.6.8.4 for additional discussion regarding the use of ACK_0.

- **Word 0, Bit 10 - ACK_N capability**

  0 = ACK_N incapable
  1 = ACK_N capable

Bit 10 specifies that the N_Port supplying these Class Service Parameters is or is not capable of support for ACK_N as a Sequence Initiator for acknowledgement of Data frames within a Sequence. As a Sequence Initiator an N_Port receives and processes ACK frames in response to Data frame transmission. ACK_N support is applicable to Class 1 and 2, but not Class 3. ACK_1 support is mandatory, whereas support for the ACK_N form of ACK is optional.

N_Port A	N_Port B	
Word 0 Bit 10	Word 1 Bit 30	N_Port A as Sequence Initiator
0	0	ACK_N not supported
0	1	ACK_N not supported
1	0	ACK_N not supported
1	1	ACK_N supported

If one N_Port (N_Port A, for example) is capable of receiving ACK_N as a Sequence Initiator (Word 0, Bit 10 = 1) and the other N_Port (N_Port B, for example) is capable of transmitting ACK_N as a Sequence Recipient (Word 1, Bit 30 = 1), then ACK_N is supported when N_Port A is the Sequence Initiator and N_Port B is the Sequence Recipient (see 20.3.2.2 and 26.4.3.3). Otherwise, ACK_N is not supported and shall not be used while N_Port A is the Sequence Initiator and N_Port B is the Sequence Recipient. ACK_N usage shall take precedence over ACK_1 but ACK_0 usage shall take precedence over ACK_N.

ACK_N capability may be asymmetrical for a single N_Port. That is, an N_Port may be capable processing ACK_N as a Sequence Initiator, but not be capable of ACK_N transmission as a Sequence Recipient. Similarly, an N_Port may be capable of transmitting ACK_N as a Sequence Recipient, but not be capable of ACK_N reception as a Sequence Initiator. ACK_N may be supported for all forms of Discard Error Policies. ACK_N support shall not be used for Process Policy with infinite buffers Exchange Error Policy. See 23.6.8.4 for additional discussion regarding the use of ACK_N.

NOTE - Acknowledging more than one Data frame with ACK_N should only be used when multiple Data frames are ready to be acknowledged. ACK_N should be transmitted immediately when a single frame is ready to be acknowledged. An unnecessary delay in acknowledging frames may lead to timeout conditions in which the Sequence Initiator has no end-to-end Credit to transmit additional frames, while the Sequence Recipient is waiting for more frames before transmitting ACK_N.

23.6.8.4 Recipient control

Recipient Control Flags (C) shall specify which functions are supported by the N_Port supplying the Service Parameters when acting as a receiver of Data frames. Recipient Control Flags specify the R cipient functions supported by the N_Port.

Word,Bits	Recipient Ctl Flags (C)
1,31	ACK_0 support
1,30	ACK_N support
1,29	X_ID interlock
1,28,27	Error policies supported
1,26	Reserved
1,25-24	Categories per Sequence
1,23-20	Reserved
1,19-16	Reserved for Fabric specific

Figure 66 - Recipient Ctl flags (C)

- **Word 1, Bit 31 - ACK_0 capability**

  0 = ACK_0 incapable
  1 = ACK_0 capable

Bit 31 specifies that the N_Port supplying these Class Service Parameters is or is not capable of support for ACK_0 as a Sequence Recipient for acknowledgement of an entire Sequence in either Discard or Process Exchange Error Policies. As a Sequence Recipient an N_Port shall support infinite buffering and be capable of transmitting ACK_0 frames in response to Data frame transmission. ACK_0 support is applicable to Class 1 and 2, but not Class 3. ACK_1 support is mandatory, whereas support for the ACK_0 form of ACK is optional.

N_Port A Word 0 Bit 11	N_Port B Word 1 Bit 31	N_Port B as Sequence Recipient:
0	0	ACK_0 not supported
0	1	ACK_0 not supported
1	0	ACK_0 not supported
1	1	ACK_0 supported

If one N_Port (N_Port A, for example) is capable of receiving ACK_0 as a Sequence Initiator (Word 0, Bit 11 = 1) and the other N_Port (N_Port B, for example) is capable of transmitting ACK_0 as a Sequence Recipient (Word 1, Bit 31 = 1), then ACK_0 may be used when N_Port A is the Sequence Initiator and N_Port B is the Sequence Recipient. Otherwise, ACK_0 shall not be supported while N_Port A is the Sequence Initiator and N_Port B is the Sequence Recipient. If ACK_0 is supported, as indicated, N_Port A shall transmit Data fram s to N_Port B with infinite buffering in effect and shall be able to receive ACK_0 for Sequences transmitted. If ACK_0 is supported, as indicated, N_Port B shall support infinit buffering and be capable of transmitting ACK_0 to acknowledge an entire Sequence (see 20.3.2.2 and 26.4.3.1).

ACK_0 capability may be asymmetrical for a single N_Port. That is, an N_Port may be capable processing ACK_0 as a Sequence Initiator, but not be capable of ACK_0 transmission as a Sequence Recipient. Similarly, an N_Port may be capable of generating ACK_0 as a Sequence Recipient, but not be capable of ACK_0 reception as a Sequence Initiator. If an N_Port sets both Word 0, bit 11 and Word 1, bit 31 to one, then it is capable of ACK_0 support as either a Sequence Initiator or a Sequence Recipient.

ACK_0 may be used for all forms of Discard Error Policies for Exchanges. ACK_0 support shall be required for Process Policy with infinite buffers Exchange Error Policy. However, any requirements regarding symmetrical or assymmetrical ACK_0 support shall be specified by the FC-4 or upper level and is outside the scope of FC-PH. For example, if an FC-4 only transmitted Video_Data from N_Port A to N_Port B (a frame buffer), then N_Port A need only support ACK_0 as a Sequence Initiator, whereas, N_Port B need only support ACK_0 as a Sequence Recipient. See 23.6.8.4 for additional discussion regarding the use of ACK_0.

However, even if ACK_0 support is indicated by both N_Ports, a Sequence Recipient shall use ACK_1 or ACK_N (ACK_CNT = 1) within an Exchange for a Sequence of more than one frame for three cases:

a) X_ID interlock,
b) in the response to a connect-request (**SOFc1**), or
c) setting the Abort Sequence Condition bits to a value other than 0 0.

- **Word 1, Bit 30 - ACK_N capability**

  0 = ACK_N incapable
  1 = ACK_N capable

Bit 30 specifies that the N_Port supplying these Class Service Parameters is or is not capable of support for ACK_N as a Sequence Recipient for acknowledgem nt of Data frames within a Sequence. As a Sequence Recipient an N_Port receives Data fram s and transmits ACK_N frames in response to Data frame reception. ACK_N support is applicable to Class 1 and 2,

but not Class 3. ACK_1 support is mandatory, whereas support for the ACK_N form of ACK is optional.

N_Port A	N_Port B	
Word 0 Bit 10	Word 1 Bit 30	N_Port B as Sequence Recipient:
0	0	ACK_N not supported
0	1	ACK_N not supported
1	0	ACK_N not supported
1	1	ACK_N supported

If one N_Port (N_Port A, for example) is capable of receiving ACK_N as a Sequence Initiator (Word 0, Bit 10 = 1) and the other N_Port (N_Port B, for example) is capable of transmitting ACK_N as a Sequence Recipient (Word 1, Bit 30 = 1), then ACK_N shall be used when N_Port A is the Sequence Initiator and N_Port B is the Sequence Recipient (see 20.3.2.2 and 26.4.3.3). Otherwise, ACK_N shall not be used while N_Port A is the Sequence Initiator and N_Port B is the Sequence Recipient.

ACK_N capability may be asymmetrical for a single N_Port. That is, an N_Port may be capable processing ACK_N as a Sequence Initiator, but not be capable of ACK_N transmission as a Sequence Recipient. Similarly, an N_Port may be capable of transmitting ACK_N as a Sequence Recipient, but not be capable of ACK_N reception as a Sequence Initiator. If an N_Port sets both Word 0, bit 10 and Word 1, bit 30 to one, then it is capable of ACK_N support as either a Sequence Initiator or a Sequence Recipient.

ACK_N may be supported for all forms of Discard Error Policies. ACK_N support shall not be used for Process Policy with infinite buffers Exchange Error Policy. See 23.6.8.3 for additional discussion regarding the use of ACK_N.

NOTE - Acknowledging more than one Data frame with ACK_N should only be used when multiple Data frames are ready to be acknowledged. ACK_N should be transmitted immediately when a single frame is ready to be acknowledged. An unnecessary delay in acknowledging frames may lead to timeout conditions in which the Sequence Initiator has no end-to-end Credit to transmit additional frames, while the Sequence Recipient is waiting for more frames before transmitting ACK_N.

- **Word 1, Bit 29 - X_ID interl ck**

  0 = X_ID interlock not required

  1 = X_ID interlock required

X_ID interlock only applies to Class 1 and Class 2. This bit indicates that the N_Port supplying this parameter requires that an interlock be used during X_ID assignment or reassignment in Class 1 and 2. In X_ID assignment or reassignment (see 25.3.2), the Sequence Initiator shall set the Recipient X_ID value to hex 'FFFF' in the first Data frame of a Sequence and the Recipient shall supply its X_ID in the ACK frame corresponding to the first Data frame of a Sequence. The Sequence Initiator shall not transmit additional frames until the corresponding ACK is received. Following reception of the ACK, the Sequence Initiator continues transmission of the Sequence using both assigned X_ID values.

- **Word 1, Bits 28-27 - Error policy supported**

  0 0 = Only discard supported
  0 1 = Reserved
  1 0 = Both discard and process supported
  1 1 = Reserved

These bits are set to specify the types of support possible for missing frame conditions processed by the Receiver N_Port. The policy used for a given Exchange shall be specified as discard or process by the Exchange Originator (see Abort Sequence Condition bits in 18.5 and 29.6.1.1).

In either **Discard or Process policy**, when a missing frame error is detected, the expected sequence count is saved in the Error SEQ_CNT field of the appropriate Sequence Status Block and a Sequence error is posted in the S_STAT field in the same Sequence Status Block for a given Exchange (OX_ID, RX_ID). Only the first error is saved.

NOTE - The error status is reported by FC-2 to FC-4.

Discard policy support specifies that the N_Port shall be able to discard Data frames received following detection of a missing frame error condition including the frame at which the error is detected.

Except for Class 3, when the missing frame condition is detected, the Sequence Recipient shall notify the Sequence Initiator using the Abort Sequence Condition bits in F_CTL on the ACK corresponding to the frame on which the error was detected. All subsequent frames continue to be discarded and Abort Sequence indicated in

the ACK frame (see 24.3.10). In one Discard Policy, only a single Sequence shall be discarded, whereas in the other Discard Policy, multiple Sequences shall be discarded (see Abort Sequence Condition bits in 18.5 and 29.6.1.1).

Process policy support specifies that the N_Port is to able to continue processing valid Data frames following a detected missing frame error condition in the normal manner, including the frame at which the error is detected (see 29.6.1.1).

- **Word 1, Bits 25-24 - Categories per Sequence**

  0 0 = 1 Category/Sequence
  0 1 = 2 Categories/Sequence
  1 0 = Reserved
  1 1 = More than 2 Categories/Sequence

The setting of bits 25-24 shall specify that the Recipient is capable of processing one, two, or more than two Information Categories (R_CTL bits 27-24) in a single Sequence. Bits 25-24 are applicable to each Class of Service since support for an individual Class may offer different capabilities in the same N_Port.

When an N_Port is acting as a Sequence Initiator, it shall restrict the number of Information Categories per Sequence based on the Sequence Recipient's capability as specified during N_Port Login. An N_Port's capability for processing Information Categories in a single Sequence may prohibit that N_Port from communicating in certain FC-4 protocols.

Each FC-4 should allow the ability to communicate using only one Information Category per Sequence but always provide the ability to communicate using multiple Information Categories per Sequence where possible, and when performance may be enhanced.

23.6.8.5 Receive Data_Field Size

- **Word 1, Bits 15-0 Receive Data_Field Size (N)**

The **Receive Data_Field Size** is a binary valu (bits 15-0) which specifies the largest Data_Field Size for an FT_1 frame (17.4) that can be received by the N_Port supplying the Service Parameters as a Sequence Recipient. Values less than 128 or greater than 2 112 are invalid. Values shall be a multiple of four bytes. An N_Port shall support a Data Field size of at least 128 bytes, however, a minimum of 256 bytes is recommended.

Class 1
In Class 1 the Receive Data_Field size represents the largest Data_Field size that an N_Port is able to receive after a Dedicated Connection is established. (The connect-request Data_Field size is specified in the Buffer-to-Buffer Receive Data_Field size in Common Service Parameters.)

Class 2 and 3
The Receive Data_Field size for Class 2 and 3 shall be equal to or less than the Buffer-to-Buffer Receive Data_Field size specified in the Common Service Parameters.

The maximum Receive Data_Field Size is specified by FC-2.

23.6.8.6 Concurrent Sequences

- **Word 2, Bits 31-16 Concurrent Sequences (L)**

Concurrent Sequences shall specify the number of Sequence Status Blocks available in the N_Port supplying the Service Parameters for tracking the progress of a Sequence as a Sequence Recipient. The maximum number of Concurrent Sequences that can be specified is 255 per N_Port as a Recipient which may be allocated across all three Classes. The total number of Concurrent Recipient Sequences which may be Open or Active across all three Classes by a single N_Port is specified in the Common Service Parameter field (see 23.6.3).

Bits
2222 1111
3210 9876

0000 0000 = Reserved

0000 0001 = 1 Sequence Status Register

....
1111 1111 = 255 Sequence Status Registers

NOTE - If each N_Port specified 255 Concurrent Sequences, then the maximum number of Open

Sequences between the two N_Ports is 510 (255 as Recipient and 255 as Initiator, for each N_Port).

Class 1

The SEQ_ID values shall range from 0 to L, inclusively, where L is the value of the Concurrent Sequence field. During a Class 1 Connection an N_Port shall support the maximum number of Concurrent Sequences specified during Login.

Class 2 and 3

The SEQ_ID values shall range from 0 to 255. In Class 2 an N_Port is allowed to respond with a P_BSY to a frame initiating a new Sequence if N_Port resources are not available.

23.6.8.7 End-to-end Credit

- **Word 2, Bits 14-0 End-to-end Credit (M)**

End-to-end Credit is the maximum number of Data frames which can be transmitted by an N_Port without receipt of accompanying ACK or Link_Response frames. The minimum value of end-to-end Credit is one. The end-to-end Credit field specified is associated with the number of buffers available for holding the Data_Field of a frame and processing the contents of that Data_Field by the N_Port supplying the Service Parameters. End-to-end Credit is not applicable to Class 3 since ACK frames are not used.

NOTE - In order to ensure frame identification integrity, end-to-end Credit is defined as a 15 bit field while sequence count is a 16 bit field. This ensures that end-to-end Credit can never exceed one-half of the maximum sequence count. Bit 15 shall be set = 0.

Values in the end-to-end Credit Field have the following meanings:

```
  Bits
111 11
432 1098 7654 3210

000 0000 0000 0000 = Reserved
000 0000 0000 0001 = 1 Buffer
.... ....
111 1111 1111 1110 = 32 766 Buffers
111 1111 1111 1111 = 32 767 Buffers
```

23.6.8.8 Open Sequences p r Exchange

- **Word 3, Bits 31-16 Open Sequences / Exchange (X)**

The value of Open Sequences per Exchange shall specify the maximum number of Sequences that can be Open at the Recipient at one time between a pair of N_Ports for one Exchange. The value of X+2 specifies the number of instances of Sequence Status that shall be maintained by the Recipient for a single Exchange in the Exchange Status Block. This value is used for Exchange and Sequence tracking. The value of X limits the link facility resources required for error detection and recovery (see clause 29). The value of X is specified in bits 23-16.

NOTE - The number of SSBs specified at X+2 to be retained in the ESB ensures that if Sequence streaming rules are followed, the ESB shall contain at least one "good" Sequence that ended normally. Another SSB position was allocated in order to allow for any race or timing conditions that might impact that "good" Sequence.

23.6.9 Vendor Version Level

Vendor Version Level is a 16-byte field which specifies FC-PH version levels for an N_Port which deviate in a Vendor-specific manner from the FC-PH version levels specified in the Common Service Parameters. If Word 1, bit 29 = 1 in the Common Service Parameters, the Vendor Version Level field contains valid information. If Word 1, bit 29 = 0, the Vendor Version Level field is not meaningful.

The first two bytes specify a code assigned to a specific vendor. The next 6 bytes specify one or more Vendor-specific versions supported by the N_Port supplying these Login Parameters. The levels may be encoded or bit-assigned. The last 8 bytes are reserved for Vendor-specific use.

23.7 F_Port Service Parameters

The first sixteen bytes of the Payload following the LS_Command Code in the Accept Sequence shall specify F_Port Service Parameters common to all three Classes. The next eight bytes shall contain the F_Port_Name of the F_Port transmitting th Service Parameters. The next eight bytes shall contain the Fabric_Name of the Fabric which controls the F_Port.. The next 48 bytes of the Payload following the Fabric_Name

shall specify three sets of Class Service Parameters as shown in figure 59. The first 16-byte group shall specify Service Parameters for Class 1. The second 16-byte group shall specify the Service Parameters for Class 2. The third 16-byte group shall specify the Service Parameters for Class 3. The fourth and fifth 16-byte groups shall be reserved.

F_Port Service Parameters in the Accept Link Service Sequence have different interpretations than Login data supplied by an N_Port.

23.7.1 F_Port Common Service Parameters

F_Port Common Service Parameters represent Parameters which are common across all Classes supported by an F_Port and are shown in figure 67.

Word	Bits 31	0
0	FC-PH Version HHHHHHHH LLLLLLLL	Buffer-to-Buffer Credit BBBBBBBB BBBBBBBB
1	Common Features rrrNrrrr rrrrrrrr	Buffer-to-Buffer Rcv Data Field Size rrrrFFFF FFFFFFFF
2	R_A_TOV tttttttt tttttttt	tttttttt tttttttt
3	E_D_TOV tttttttt tttttttt	tttttttt tttttttt

Figure 67 - F_Port Common Service Parameters

23.7.1.1 FC-PH version

The FC-PH version field provides a method of determining the version of FC-PH which an F_Port shall be capable of supporting. The FC-PH field is divided into two one-byte fields. Table 102 indicates the hexadecimal values for the low and high FC-PH version levels in word 0 bits 31-24, and bits 23-16.

Table 102 - FC-PH Version - F_Port

Hex value	Version level
00	None
01 - 05	Reserved
06	4.0
07	4.1
08	4.2
09	4.3
others	Reserved

A version of FC-PH shall be assigned a new binary value when its implementation is not compatible with the previous version specified. An F_Port shall support all versions between the lowest and the highest levels specified.

- **Word 0, Bits 31 - 24 - Highest version supported (H)**

The binary value of bits 31-24 are encoded to represent the highest or most recent version of FC-PH which the F_Port shall be capable of supporting.

- **Word 0, Bits 23 - 16 - Lowest version supported (L)**

The binary value of bits 23-16 are encoded to represent the lowest or earliest version of FC-PH that the F_Port shall be capable of supporting.

Where there is overlap in the FC-PH versions supported, each F_Port shall operate in the highest FC-PH version mutually supported.

23.7.1.2 Buffer-to-buffer (F_P rt) Credit

The buffer-to-buffer Credit (B) field (Word 0, bits 15-0) specified is associated with the number of buffers available for holding Class 1 connect-request, Class 2, or Class 3 frames received from the N_Port.

The buffer-to-buffer Credit (B) shall be a single value which represents the total buffer-to-buffer Credit available for all Classes. An N_Port or F_Port tracks buffer-to-buffer Credit as a single entity for all frames subject to buffer-to-buffer flow control (see clause 26).

Values in the buffer-to-buffer Credit field have the same format as in the end-to-end Credit Class Service Parameters.

23.7.1.3 Common features

- **Word 1, Bit 28 - N_Port/F_Port (N)**

  0 = N_Port
  1 = F_Port

Word 1 bit 28 shall be set = 1.

23.7.1.4 Buffer-to-buffer Data_Field size

- **Word 1, Bits 15-0 Buffer-to-buffer Receive Size (F)**

The **buffer-to-buffer Receive Data_Field Size** is a binary value (bits 15-0) which specifies the largest Data_Field Size for an FT_1 frame (17.4) that can be received by the F_Port supplying the Service Parameters for:

– a connect-request (**SOF_{c1}**),
– a Class 2 Data frame, or
– a Class 3 Data frame.

Values less than 128 or greater than 2 112 are invalid. Values shall be a multiple of four bytes. An F_Port shall support a Data Field size of at least 128 bytes, however, a minimum of 256 bytes is recommended.

23.7.1.5 E_D_TOV

The E_D_TOV value shall be specified as a count of 1 ms increments. Therefore, a value of hex '0000000A' specifies a time period of 10 ms.

23.7.1.6 R_A_TOV

The R_A_TOV value shall be specified as a count of 1 ms increments. Therefore, a value of hex '0000000A' specifies a time period of 10 ms.

23.7.2 F_Port_Name

The F_Port_Name is an eight byte field. The format of the F_Port_Name is specified in 19.3. Each F_Port shall provide a unique F_Port_Name within the address domain of the Fabric and associated N_Ports. Bits 63-60 specify the format of the F_Port_Name. The formats are summarized in tables 43 and 44 in 19.3.

23.7.3 Fabric_Name

The Fabric_Name is an eight byte field. The format of the Fabric_Name is specified in 19.3. Each Fabric shall provide a unique Fabric_Name. Bits 63-60 specify the format of the Fabric_Name. The formats are summarized in tables 43 and 44 in 19.3.

23.7.4 F_Port Class Service Parameters

The 48 bytes of the Payload following the Fabric_Name shall contain 48 bytes to specify three sets of Class Service Parameters as shown in figure 59. The first sixteen-byte group specifies Service Parameters for Class 1. The second sixteen-byte group specifies the Service Parameters for Class 2 and the third sixteen-byte group specifies the Service Parameters for Class 3.

Within each group of 16-byte F_Port Class Service Parameters, fields are identical to those shown in figure 62.

23.7.4.1 Class validity (V)

- **Word 0, Bit 31 - Class validity**

  0 = Invalid (Class not supported)
  1 = Valid (Class supported)

The Class validity bit indicates whether this Class is supported. If the Class validity bit is zero, it indicates that this set of sixteen bytes shall be ignored. If the Class validity bit is one, it indicates that this Class is supported. The Class is identified based on Class 1 = first sixteen-byte group, Class 2 = second sixteen-

byte group, and Class 3 = third sixteen-byte group.

23.7.4.2 Service options

Service Options (E) shall specify Class of Service capabilities supported or required by the Fabric supplying the Service Parameters.

- **Word 0, Bit 30 - Intermix mode**

  0 = Intermix not supported
  1 = Intermix supported

Class 1

All Fabrics supporting Class 1 shall support Exclusive Connections. An N_Port supporting Exclusive Connections may only transmit and receive frames from the N_Port to which an existing Class 1 Connection is established. Exclusive Connections require that the Fabric transmit an F_BSY frame, as appropriate, in response to frames issued by a third N_Port targeted for one of the two N_Ports engaged in a Class 1 Connection.

An Intermixed Dedicated Connection specifies that the Fabric may insert or extract Class 2 or Class 3 frames while a Class 1 Connection is established. Support for Intermix is optional by both an N_Port and a Fabric. If the N_Port supports intermixing of Class 2 and 3 frames during a Class 1 Connection, it shall request Fabric support for Intermix during Login with the Fabric. See 22.4 for more discussion on Intermix.

If both the N_Port and the F_Port have indicated support for Intermix Connections, then Intermix mode shall be functional.

N_Port	F_Port	
0	0	Neither supports
0	1	Fabric is capable of supporting Intermix not functional
1	0	N_Port support requested, Fabric does not support
1	1	N_Port requested, Fabric agrees, Intermix is functional

Class 2 and 3

Word 0 bit 30 is reserved for Class 2 and 3.

- **Word 0, Bits 29-28 - Stacked connect-requests**

Support for stacked connect-requests is optional in both a Fabric and an N_Port. Both an N_Port's and F_Port's behavior change if stacked connect-requests are functional.

Stacked connect-requests require Intermix mode support. Therefore, an F_Port shall only indicate support for either mode of stacked connect-requests if Intermix support has also been requested and is functional. Support for stacked connect-requests allows a Fabric to process that connect-request while an N_Port is Connected to another N_Port in order to reduce connect-request latency.

Due to the timing relationships involved, there are two methods of implementing stacked connect-requests by a Fabric:

a) transparent mode, or
b) lock-down mode.

In transparent mode, when the **SOFc1** Data frame is delivered to the destination N_Port, the return path of the bi-directional circuit is established in the same manner as Exclusive Dedicated Connections. This means that the destination N_Port of the **SOFc1** is able to transmit Data frames immediately following transmission of the ACK frame in response to the **SOFc1** frame.

In lock-down mode, when the **SOFc1** Data frame is delivered to the destination N_Port, the return path of the bi-directional circuit is not necessarily established to the source N_Port of the **SOFc1**. Therefore, the **SOFc1** Data frame shall also set F_CTL bit 8 = 1 (Unidirectional Transmit) in order to inhibit the destination N_Port of the **SOFc1** from sending any Data frames after the ACK frame is transmitted in response to the connect-request (see 28.5.2 for more information on stacked connect-requests and 18.5 for more information on F_CTL bit 8).

Bit 29 is a request by the N_Port for transparent stacked connect-requests. Bit 28 is a request by the N_Port for lock-down stacked connect-requests. An N_Port may request either one or both modes. The Fabric shall support transparent mode or lock-down mode but not both modes.

W rd 0, bit 29 Transparent mode - stacked connect-request

0 = Transparent mode not supported
1 = Transparent mode functional

Class 1

When an N_Port performs Login with a Fabric, it shall request support for transparent stacked connect-requests by specifying bit 29 = 1. Bit 29 = 1 is only meaningful if Intermix is functional. If the Accept reply from the Fabric specifies bit 29 = 1, then both the N_Port and Fabric have agreed that transparent stacked connect-requests are functional.

The following set of values specifies the meaning of the combination of Word 0, bit 29:

N_Port	F_Port	
0	0	Neither supports
0	1	Fabric is capable of supporting, transparent mode stacked connect-requests not functional
1	0	N_Port support requested, Fabric does not support
1	1	N_Port requested, Fabric agrees, transparent mode stacked connect-requests are supported

Class 2 and 3

Word 0, bit 29 is reserved.

Word 0, bit 28 Lock-down mode - stacked connect-request

0 = Lock-down mode not supported
1 = Lock-down mode supported

Class 1

When an N_Port performs Login with a Fabric, it shall request support for lock-down stacked connect-requests by specifying bit 28 = 1. Bit 28 = 1 is only meaningful if Intermix is functional. If the Accept reply from the Fabric specifies bit 28 = 1, then both the N_Port and Fabric have agreed that lock-down stacked connect-requests are functional.

The following set of values specifies the meaning of the combination of Word 0, bit 28:

N_Port	F_Port	
0	0	Neither supports
0	1	Fabric is capable of supporting, lock-down mode stacked connect-requests not functional
1	0	N_Port support requested, Fabric does not support
1	1	N_Port requested, Fabric agrees, lock-down mode stacked connect-requests are functional

Class 2 and 3

Word 0, bit 28 is reserved.

- **Word 0, Bit 27 - Sequential delivery**

  0 = Out of order delivery
  1 = Sequential delivery functional

If bit 27 is set to one by an N_Port, then it is requesting that all frames delivered to the N_Port requesting this function be delivered in the same order in which the frames were transmitted by the source N_Port. If bit 27 is set to one by the F_Port, then the F_Port is indicating that it shall deliver Class 2 and 3 frames in the same order as transmitted from any one N_Port.

A Fabric supporting the sequential delivery feature routes Class 2 and 3 frames via a fixed route through the Fabric in order to provide in-order frame delivery. This feature does not imply any other alteration to the normal class of service functions such as F_BSY.

If the N_Port has requested this support and the Fabric responds with bit 27 = 1, then in-order delivery is being supported by the Fabric. If the N_Port has not requested this support and the Fabric responds with bit 27 = 1, then in-order delivery is functional by the Fabric as default operation.

Class 1

Bit 27 has no meaning in Class 1 since the Fabric is required to deliver frames in the order transmitted based on class of service.

Class 2

In Class 2, if bit 27 = 1, the Fabric guarantees the order of delivery of both Data and Link_Control frames to correspond to the order in which the frames are transmitted.

Class 3

In Class 3, if bit 27 = 1, the Fabric guarantees the order of delivery of Data frames to correspond to the order in which the frames are transmitted.

```
N_Port  F_Port
 0       0   Neither supports
 0       1   Sequential delivery by the
             Fabric is functional
 1       0   N_Port support requested,
             Fabric does not support
 1       1   N_Port requested, Fabric agrees,
             Sequential delivery
               is functional
```

23.7.4.3 Initiator control

This field is not meaningful.

23.7.4.4 Recipient control

- **Word 1, Bits 19-16 Reserved for Fabric specific use**

These bits are reserved for specific use for Fabric features.

23.7.4.5 Receive Data_Field Size

This field is not meaningful.

23.7.4.6 Concurrent Sequences

This field is not meaningful.

23.7.4.7 N_Port End-to-end Credit.

This field is not meaningful.

23.7.4.8 Open Sequences per Exchange

This field is not meaningful.

23.8 Procedure to estimate end-to-end Credit

23.8.1 Introduction

An estimate of the minimum end-to-end Credit between an N_Port pair for a given distance helps achieve the maximum bandwidth utilization of the channel, by continuously streaming data. The procedure to estimate end-to-end Credit is defined to accomplish this purpose.

Link Service Sequences which support this procedure are optional. This procedure shall be performed after Login between this N_Port pair. Login determines a number of Service Parameters such as the maximum frame size that can be received by each N_Port.

Applicability
The procedure is applicable to both Class 1 and Class 2. In Class 2, the procedure and the continuous streaming function may also be limited by the buffer-to-buffer Credit.

Users
The procedure shall be invoked by the Link Service support of the source N_Port and responded to by the Link Service support of the destination N_Port. Since the Extended Link Service requests used to perform this procedure are optional, LS_RJT may be received to any request with a reason code of Command Not Supported

Prerequisite
To perform this procedure for Class 1 or Class 1 Intermix, a Class 1 Connection shall have been established before the procedure is performed.

23.8.2 Procedure steps

This procedure is optional and consists of following three request Sequences.

a) Establish Streaming Sequence

b) Estimate Credit Sequence

c) Advise Credit Sequence

The procedure is illustrated with these request Sequences and their respective reply Sequences in figure 68.

The procedure shall be performed for Class 1 or Class 2 with respective delimiters, as defined in clause 22.

23.8.2.1 Establish Streaming Sequence

This Sequence shall be used to obtain an end-to-end Credit large enough to perform continuous streaming from a source N_Port to a destination N_Port. This Sequence provides an opportunity for the destination N_Port to communicate the maximum end-to-end Credit it can provide for the purposes of streaming. This temporary allocation is termed Streaming Credit (L).

This Sequence shall be used between an N_Port pair after the N_Port pair have logged in with each other. This Sequence shall be initiated as a new Exchange. The Sequence shall be initiated by the Originator of the Exchange.

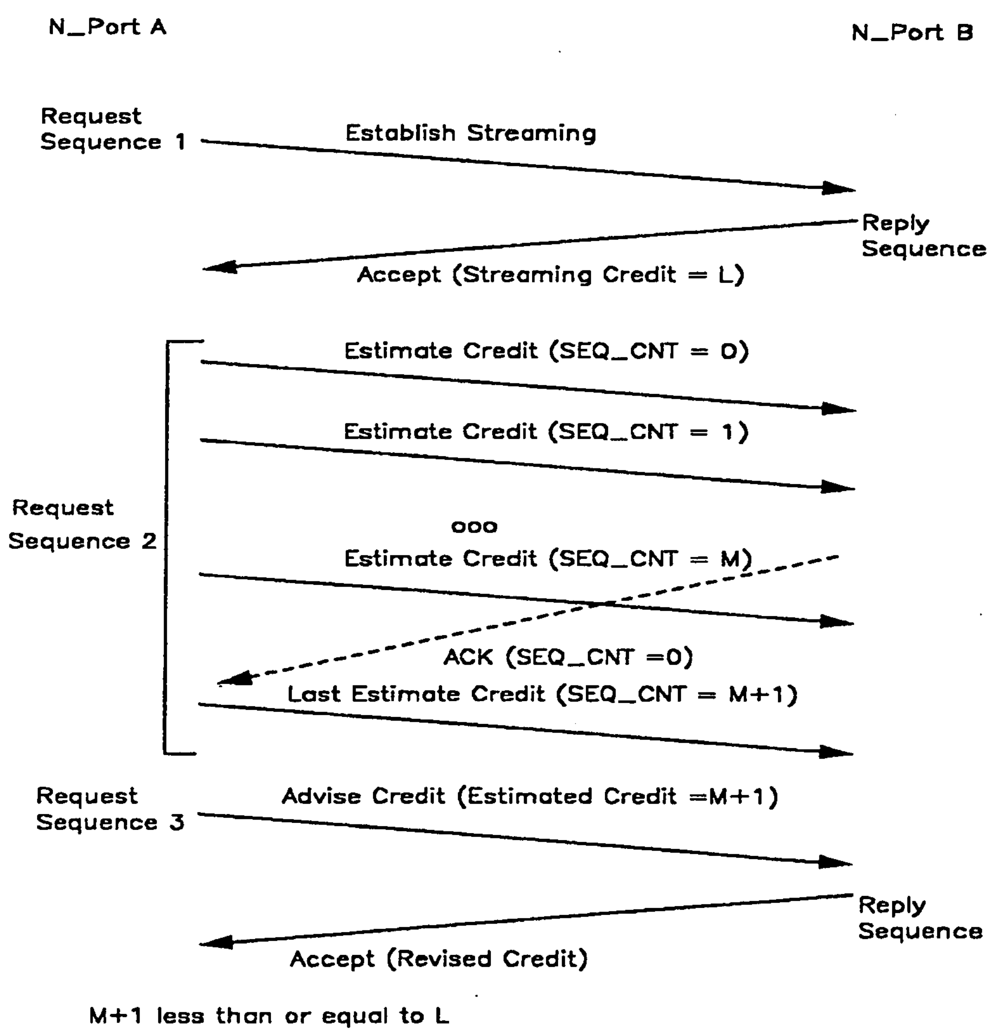


Figure 68 - Pr cedure to estimate end-to-end Credit

a) The source shall transmit the Establish Streaming (ESTS) frame.

b) The destination shall reply with an ACC frame.

c) The Class Validity bit = 1 shall identify the Class which is requesting streaming Credit.

d) The Payload of ACC shall have the same format as the Service Parameters for N_Port Login. The Payload shall contain Streaming Credit (L) allocated in the end-to-end Credit field of the appropriate Class of the Service Parameters for which the Estimate Credit procedure is being performed (word 2, bits 14-0 of each class group). The other fields shall be ignored by the receiver.

23.8.2.2 Estimate Credit Sequence

This Sequence shall be performed immediately following the completion of the Establish Streaming Sequence.

a) The source N_Port shall stream Estimate Credit (ESTC) frames consecutively until it receives the first ACK (ACK_1 or ACK_N) from the destination N_Port which shall set the Abort Sequence bits (F_CTL bits 5-4) = 1 0. The source shall not exceed the Streaming Credit obtained during the Establish Streaming Sequence.

b) If the source does not receive ACK (ACK_1 or ACK_N) after it has reached the limit imposed by the Streaming Credit value, it shall stop streaming and wait for the first ACK to be received with the Abort Sequence bits (F_CTL bits 5-4) = 1 0.

c) The size of the Data Field of the ESTC frame shall be the normal size frames transmitted by an FC-4 based on the Service Parameters from N_Port Login.

d) The Payload shall contain valid data bytes.

e) The SEQ_CNT shall follow the normal rules for Sequence transmission.

f) The destination N_Port shall respond with ACK for Data frames received.

g) If the highest SEQ_CNT transmitted by the source N_Port at the time it receives the first ACK is M, the number of outstanding frames (i.e., Credit estimated for continuous streaming) shall equal M+1. If ACK is received within the Streaming Credit limit (L > M), this value of M+1 represents the minimum Credit required to utilize the maximum bandwidth of the Fibre. If the ACK is received after reaching the Streaming Credit limit (L), this value is less than the optimal Credit required to utilize the maximum bandwidth of the Fibre.

h) The source N_Port shall follow all the rules in closing the Sequence, by sending the last Data frame of the Sequence and waiting for corresponding ACK to be received.

23.8.2.3 Advise Credit Sequence

This Sequence shall be performed immediately following completion of the Estimate Credit Sequence. The source N_Port which performed the Estimate Credit Sequence shall advise the destination N_Port of the Estimated Credit in ADVC Data Field. The destination N_Port shall reply using an ACC frame, with a revised end-to-end Credit value in its Payload. This value is determined by the destination N_Port based on its buffering scheme, buffer management, buffer availability and N_Port processing time. This is the final value to be used by the source N_Port for revised end-to-end Credit.

This Sequence provides a complementary function to Login. In contrast to the Login frame, the ADVC frame contains the end-to-end Credit it would like to be allocated for continuous streaming.

If the Estimated Credit (M+1) is less than or equal to the Streaming Credit (L), the destination may choose to reallocate the estimated end-to-end Credit. If the Streaming Credit (L) is smaller than needed for continuous streaming, the source N_Port is bound to run short of end-to-end Credit and the source N_Port may advise that value as the Estimated Credit.

a) The source N_Port shall transmit Advise Credit frame with the Estimated Credit (M+1).

b) The Payload of the ADVC shall have the same format as the Service Parameters for Login. The Payload shall contain the Estimated Credit (M+1) in end-to-end Credit field of the appropriate Class of the Service Parameters. The appropriate Class is identified by the Class Validity bit = 1. The other fields shall be ignored by the receiver.

c) The destination N_Port shall determine the revised end-to-end Credit value. The destination shall determine the value based on its

buffer management, buffer availability and port processing time and may add a factor to the Estimated Credit value. This is the final value to be used by the source N_Port for end-to-end Credit.

d) The destination N_Port replies with an ACC frame which successfully completes the Protocol. The ACC Sequence shall contain the end-to-end Credit allocated to the source N_Port. The Payload of ACC shall have the same format as the Service Parameters for Login. The Payload shall contain the final end-to-end Credit in end-to-end Credit field of the appropriate Class of the Service Parameters. The appropriate Class is identified by Class Validity bit = 1. The other fields shall be ignored by the receiver.

Asymmetry
Since the maximum frame size is permitted to be unequal in forward and reverse directions, the Estimate Credit procedure may be performed separately for each dir ction of transf r. Credit modification applies only to the direction of the transfer of Estimate Credit frames.

Frequency
The Estimate Credit procedure provides an approximation of the distance involved on a single path. If there are concerns that in a Fabric in which the length (and time) of the paths assigned can vary, the procedure may be repeated several times to improve the likelihood that the Estimated end-to-end Credit value is valid.

Alternatively, a source may accept the Estimated end-to-end Credit value. If, at a later time, data transfers are unable to stream continuously, the source may re-initiate the Estimate Credit Procedure, or arbitrarily request an increase in Estimated end-to-end Credit by using an ADVC Link request Sequence.

24 Exchang , Sequence, and sequence count management

24.1 Introduction

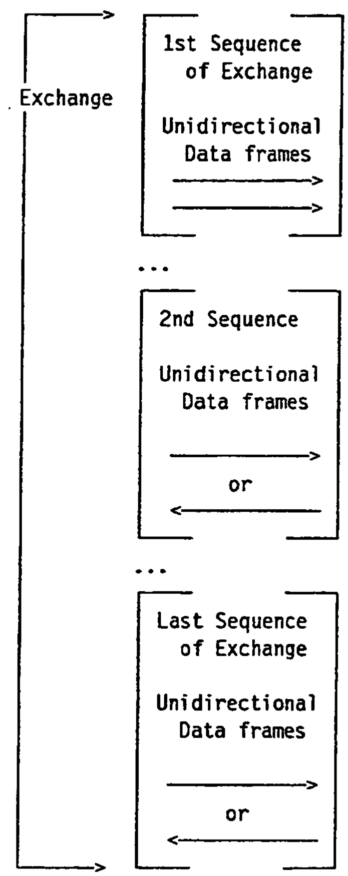
An Exchange is the fundamental mechanism for coordinating the interchange of information and data between two N_Ports or nodes. All Data transmission shall be part of an Exchange. This clause discusses Exchanges between N_Ports. FC-PH does not address the means to manage Exchanges across multiple N_Ports within a Node.

Data frame transfer
Transfer of information between two N_Ports is based on transmission of a Data frame by a source N_Port with an ACK response frame from the destination N_Port receiving the Data frame to acknowledge Data frame delivery as appropriate (Class 1 and 2).

Sequence
A Sequence is a set of one or more related Data frames transmitted unidirectionally from one N_Port to another N_Port within an Exchange. In Class 1 and 2, an ACK_1 frame is transmitted in response to each Data frame, an ACK_N frame is transmitted for N Data frames, or a single ACK_0 is transmitted for all Data frames of a Sequence. A Sequence is assigned a SEQ_ID (Sequence_Identifier) by the Sequence Initiator. A Sequence shall only be initiated when an N_Port holds the Sequence Initiative for a given Exchange.

Streamed Sequences
FC-PH allows an N_Port to initiate a new Sequence for the same Exchange following transmission of the last Data frame of a Sequence before receiving the final ACK (**EOFt**, **EOFdt**) for the previous Sequence in Class 1 and 2, or before R_A_TOV has expired for all frames of a Class 3 Sequence (i.e., streamed Sequences). See 18.6 for more information regarding the assignment of SEQ_IDs for additional rules when streaming Sequences.

Sequence count
Each frame within a Sequence contains a sequence count (SEQ_CNT) which represents the sequential number of each Data frame within one or multiple Sequences transmitted by an Exchange Originator or Responder. An ACK response frame contains a sequence count which is set equal to the Data frame sequence count to which it is responding (Class 1 and 2).



... indicates time delay

Figure 69 - Exchange - Sequence relationship

Exchange
An Exchange is a set of one or more related Sequences. Sequences for the same Exchange may flow in the same or opposite direction between a pair of N_Ports but not simultaneously (i.e. Data flows in one direction at a time within an Exchange for a single N_Port pair). Therefore, an Exchange may be unidirectional or bidirectional. Within a single Exchange only one Sequence shall be Active at any given time for a single initiating N_Port. That is, a Sequence Initiator shall complete transmission of Data frames for a Sequence of an Exchange before initiating another Sequence for the same Exchange. In Class 1, an Exchange may span multiple or successive Dedicated Connections at Sequence boundaries.

An Exchange may utilize multiple Classes of Service. Separate Sequences for the same Exchange may be transmitted in both Class 1 and 2. Class 3 Sequences shall not be transmitted in the same Exchange with Class 1 or 2 Sequences. A Sequence Initiator shall not stream Sequences that are in different Classes of Service.

Note - In Class 2, when Sequences are streamed, a Recipient N_Port may see multiple Active Sequences for the same Initiator because of out of order delivery.

The Sequence Initiator shall use continuously increasing SEQ_CNT if Sequences are streamed. If the Sequence Initiator does not stream Sequences, it may also use continuously increasing SEQ_CNT to allow the Sequence Recipient to track delivery order.

In the Discard multiple Exchange Policy, the Sequence Recipient shall deliver consecutive Sequences within an Exchange in the order transmitted. The Sequence Recipient shall preserve transmission order from one Sequence to the next even if the Sequence Initiator does not use continuously increasing SEQ_CNT. Should frames arrive out of order, the Sequence Recipient may delay transmission of the last ACK until the order is re-established.

An Exchange is assigned an Originator Exchange_ID (OX_ID) by the Originator and a Responder Exchange_ID (RX_ID) by the Responder facilities within the N_Ports specified by the N_Port Identifiers involved in the Exchange. When an Exchange is originated, there is a binding of resources in both the Originator and Responder.

An Exchange Status Block exists throughout the life of an Exchange and is located by using one or more fields of the Sequence_Qualifier, such as an N_Port's X_ID. Since an Exchange utilizes link resources, an N_Port may choose to invalidate and, subsequently, reassign its X_ID value at certain Sequence boundaries during an Exchange. An N_Port indicates its requirement to reassign X_ID values through a Service Parameter during Login.

The procedure for invalidating and, subsequently, reassigning an N_Port's X_ID uses F_CTL bits in the Frame_Header. X_IDs may be invalidated at the end of a Sequence and new X_ID values reassigned at the start of the next Sequence according to the procedure discussed in 25.3.1 and 25.3.2. When the X_ID value has been invalidated, an N_Port locates its Exchange Status Block by using an Association Header (see 19.4) until a new X_ID is assigned.

Sequence Initiative
The Exchange Originator is the Initiator of the first Sequence of the Exchange and holds the initiative to transmit Sequences. At the end of each Sequence of the Exchange, the Initiator of the Sequence may transfer the initiative to transmit the next Sequence to the other N_Port, or it may retain the initiative to transmit the next Sequence.

24.2 Applicability

Class 1 and 2:

For Class 1 and 2, FC-2 manages:

- Activation and deactivation of Exchanges,
- Initiation and termination of Sequences,
- Assignment and reassignment of X_IDs,
- Sequence Initiative,
- Assignment of SEQ_IDs,
- Segmentation and Reassembly,
- Sequences,
- Sequence count of frames,
- Detection of frame Sequence errors, and
- Notification of frame Sequence errors.

For Class 1, the Sequence Initiator shall assign SEQ_IDs from 0 to L where L is the number of Concurrent Sequences supplied by the Recipient N_Port.

NOTE - The Sequence Recipient may assign the SEQ_ID directly to a Recipient Sequence Status Block on an indexed basis.

A Class 1 Sequence requires a Class 1 Dedicated Connection for the duration of the Sequence. When the Connection is terminated, all Open Class 1 Sequences are terminated. A Sequence which has been abnormally terminated is left in an indeterminate state at the Upper Level Protocol.

For Class 2, the Sequence Initiator shall assign SEQ_IDs from 0 to 255. The Sequence Recipient assigns the SEQ_ID to an available Recipient Sequence Status Block.

Class 3:

For Class 3, the Sequence Initiator shall assign SEQ_IDs from 0 to 255. FC-2 manages the same functions as Class 1 and 2 excluding notification of frame Sequence errors to the Sequence Initiator.

24.3 Summary rules

These rules summarize the guidelines for Exchange and Sequence management. The sections that follow elaborate on these summary rules and take precedence over these guidelines.

24.3.1 Exchange management

a) Over the life of an Exchange, the Sequence Recipient shall deliver Data to FC-4 or an upper level on a Sequence basis.

b) In the Discard multiple Sequences Error Policy, each Sequence shall be delivered in the order in which the Sequence was transmitted relative to other Sequences transmitted for the Exchange.

c) In the Discard multiple Sequences Error Policy, a Sequence shall be deliverable if the Sequence completes normally and the previous Sequence, if any, is deliverable.

d) In the Discard multipl Sequences with retransmission Error Policy, each Sequence shall be delivered in the order in which the Sequence was transmitted relative to other Sequences transmitted for the Exchange.

e) In the Discard multiple Sequences with retransmission Error Policy, a Sequence shall be deliverable if the Sequence completes normally and the previous Sequence, if any, is deliverable.

f) In the Discard a single Sequence Error Policy, each Sequence shall be delivered in the order in which the Sequence was received relative to other Sequences received for the Exchange.

g) In the Discard a single Sequence Error Policy, a Sequence shall be deliverable if the Sequence completes normally without regard to the deliverability of other Sequences within the same Exchange.

h) In all Discard policies, a Sequence is complete with regard to Data content if all valid Data frames for the Sequence were received without rejectable errors being detected.

i) In Process policy with infinite buffers, a Sequence is complete with regard to Data content if at least the first and last Data frames were received as valid frames without rejectable errors being detected.

j) The ordering relationship and deliverability of Sequences between two separate Exchanges is outside the scope of FC-PH. Certain specific cases of Basic Link Service and Extended Link Service do, however, specify collision cases such as FLOGI, PLOGI, ABTX, and RSI.

24.3.2 Exchange origination

a) An Exchange being originated for Extended Link Services before Login is complete or for the purpose of Login shall follow default Login parameters and special Extended Link Services rules specified in 21.1.1 and 21.1.2 as well as clause 23.

b) A new Exchange, other than Extended Link Services, may be originated if three conditions are met:

1) The originating N_Port has performed Login with the destination N_Port.

2) The originating N_Port has an Originator_Exchange_Identifier (OX_ID) and Exchange resources available for use.

3) The originating N_Port is able to initiate a new Sequence.

c) Each frame within the first Sequence of an Exchange shall set the First_Sequence F_CTL bit to one.

d) The first frame of the first Sequence of the Exchange shall specify the Error Policy for the Exchange in F_CTL bits 5-4 of the Frame_Header. The Exchange Error Policy shall be consistent with the error policies supported by both the Originator and Responder.

e) The Originator shall transmit the first Data frame of the Exchange with its assigned OX_ID and an unassigned RX_ID of hex 'FFFF'.

— If X_ID interlock is required by the Responder (Login), the Originator (and Sequence Initiator) shall not transmit additional Data frames for this Exchange until the ACK to the first frame of the Exchange is received. The RX_ID in the ACK frame shall be used in subsequent frames of the Exchange.

— If X_ID interlock is not required by the Responder (Login), the Originator may transmit additional frames of the Sequence. In Class 1 and 2, the Responder shall return its X_ID no later than in the ACK corresponding to the last Data frame of the Sequence. The next Sequence of the Exchange shall contain both the OX_ID and RX_ID assigned in the previous Sequence.

f) In Class 1 and 2, the Sequence Initiator shall receive at least one ACK from the Recipient before the Initiator attempts to initiate subsequent Sequences for the Exchange.

g) The rules specified in Sequence initiation and termination specify the method for assigning X_IDs.

h) The rules for reassigning X_IDs are specified in 25.3.2.

24.3.3 Sequence delimiters

For a more complete description of Data frame and Link_Control delimiters see table 48 and table 50. The following rules summarize the management of frame delimiters within a Sequence:

a) A Sequence shall be initiated by transmitting the first frame with an **SOFix**, or **SOFc1**.

b) Intermediate frames within a Sequence shall be transmitted with **SOFnx** and **EOFn**.

c) The Sequence shall be complete when an **EOFt** (or **EOFdt**) has been transmitted or received for the appropriate SEQ_ID and all previous Data frames and ACKs (if any) have been accounted for by the Initiator and Recipient, respectively.

24.3.4 Sequence initiation

a) A new Sequence may be initiated if three conditions are met:

1) The initiating N_Port holds the initiative to transmit (Sequence Initiative).

2) The initiating N_Port has a SEQ_ID available for use.

3) The total number of Active Sequences initiated by the initiating N_Port with the Recipient N_Port does not exceed any of the following:

— total concurrent Sequences (see 23.6.3.5),

— concurrent Sequences per Class (see 23.6.8.6),

— Open Sequences per Exchange (see 23.6.8.8).

b) The first Data frame of the Sequence shall be started by an **SOFix** (or an **SOFc1** for the first frame establishing a Class 1 Connection).

c) The Sequence Initiator shall assign a SEQ_ID which is unique between the Initiator and Recipient N_Port Identifier pair while the Sequence is Open. The SEQ_ID shall not match the last SEQ_ID transmitted by the Sequence Initiator for this Exchange for the current Sequence Initiative.

For streamed Sequences for the same Exchange, the Sequence Initiator shall use X+1 different subsequent SEQ_IDs where X is the number of Open Sequences per Exchange so that the Exchange Status Block contains status of the last deliverable Sequence.

d) The Sequence Initiator shall not initiate the (X+1)th streamed Sequence until the first Sequence status is known. For example, if X=3 and the Sequence Initiator transmits SEQ_ID = 3, then 4, then 7, it shall not initiate another Sequence for the same Exchange until it resolves the completion status of SEQ_ID = 3, regardless of the completion status of SEQ_ID = 4 or 7.

e) The Sequence_Qualifier shall be unique until an Open Sequence is ended normally or until the Recovery_Qualifier is determined by the Abort Sequence Protocol (ABTS).

f) In Class 2 and 3 each Data frame of the Sequence shall be limited in size to the lesser of the F_Port and destination N_Port capabilities as specified by Login.

g) In Class 1 the connect-request frame shall be limited in size to the lesser of the F_Port and destination N_Port capabilities as specified by Login.

h) The Sequence Recipient shall supply the status of the Sequence in the format of the Sequence Status Block defined in 24.8.2 in response to a Read Sequence Status Extended Link Service request while the Sequence is Active. Sequence status shall be associated with the Exchange in which the Sequence is being transmitted.

i) Frame transmission shall follow Flow Control Rules as specified in clause 26 and Connection rules as specified in clause 28.

24.3.5 Sequence management

The Sequence Recipient and the Sequence Initiator should verify that frames received for a Sequence adhere to the items listed. If the Sequence Recipient determines that one of the following conditions is not met in Class 1 or Class 2, it shall transmit a P_RJT. If the Sequence Initiator determines that one of the following conditions is not met, it shall abort the Sequence (Abort Sequence Protocol).

a) Each frame shall contain the assigned SEQ_ID and assigned or reassigned OX_ID and RX_ID values.

b) Hex 'FFFF' shall be used for unassigned X_ID values.

c) Each frame shall indicate the Exchange Context.

d) Each frame shall indicate the Sequence Context.

e) Each frame shall contain a SEQ_CNT which follows the Sequenc Count rules as defined (see 24.3.6).

f) Frame transmission shall follow Flow Control Rules as specified in clause 26.

g) The Data Field size of each frame of the Sequence shall be less than or equal to:

1) the maximum size specified by the Fabric (**SOFc1**, Class 2, or 3), if present, or

2) the maximum size specified by the destination N_Port in the Service Parameters defined during Login, whichever is smaller.

h) A Sequence shall be transmitted in one Class.

i) Each Data frame in a Sequence shall be transmitted within an E_D_TOV timeout period of the previous Data frame transmitted within the same Sequence. Otherwise, a Sequence timeout shall be detected.

24.3.6 Sequence count

Within a Data frame Sequence, SEQ_CNT is used to identify each Data frame for verification of delivery and transmission order. The following rules specify the sequence count of each frame of a Sequence:

a) The sequence count of the first Data frame of the first Sequence of the Exchange transmitted by either the Originator or Responder shall be binary zero.

b) The sequence count of each subsequent Data frame within the Sequence shall be incremented by one from the previous Data frame.

c) The sequence count of the first Data frame of a streamed Sequence shall be incremented by one from the last Data frame of the previous Sequence.

d) The sequence count of the first Data frame of a non-streamed Sequence may be incremented by one from the last Data frame of the previous Sequence or may be binary zero.

e) The sequence count of each Link_Response shall be set to the sequence count of the Data frame to which it is responding (Class 1 and 2).

f) The sequence count of each ACK_1 frame shall be set to the sequence count of the Data frame to which it is responding (Class 1 and 2). See 26.4.3.2 for ACK_1 rules.

g) The sequence count of each ACK_N frame shall be set to the high st sequence count of a series of Data frames being acknowledged (Class 1 and 2). See 26.4.3.3 for ACK_N rules.

h) The sequence count of each ACK_0 frame shall be set to the sequence count of the last Data frame transmitted (End_Sequence = 1) for the Sequence (Class 1 and 2). See 26.4.3.1 for ACK_0 rules.

i) If infinite buffers and ACK_0 are being used for Sequences in which the SEQ_CNT is allowed to wrap, frame uniqueness is not being ensured (see 23.6.8.7 and 26.4.3.1 for ACK_0 rules).

j) The sequence count shall only wrap to (but not including) an unacknowledged frame in Class 2.

24.3.7 Normal ACK processing

a) Based on N_Port Login parameters (Initiator support indicated in Initiator Control and Recipient support in Recipient Control in Class Service Parameters), if both N_Ports support multiple ACK forms, ACK_0 usage shall take precedence over ACK_N and ACK_N usage shall take precedence over ACK_1. ACK_1 shall be the default if no other ACK form is supported. ACK_0 or ACK_N use may be asymmetrical between two N_Ports (see 23.6.8.3 and 23.6.8.4).

b) Mixing ACK forms in a Sequence is not allowed.

c) ACK_0 may be used for both Discard and Process Error Policies. A single ACK_0 per Sequence shall be used to indicate successful Sequence delivery or to set Abort Sequence Condition bits to a value other than 0 0. An additional ACK_0 shall be used within a Sequence to

1) perform X_ID interlock or
2) respond a connect-request

ACK_0 shall not participate in end-to-end Credit management.

d) ACK frames may be transmitted in the order in which the Data frames are processed and need not be transmitted in SEQ_CNT order, however, the History bit (bit 16) of the Parameter Field shall indicate transmission status of previous ACK frame transmission for the current Sequence.

e) The final ACK (**EOFt**) of a Sequence shall be transmitted according to the rules for normal Sequence completion (see 24.3.8) in the absence of detected errors.

f) ACK_N frames should be transmitted in order to acknowledge the processing of multiple frames in the same contiguous SEQ_CNT range as soon as the processing is complete. Accumulating information and delaying transmission of an ACK_N frame exposes the Sequence Initiator to loss of end-to-end Credit.

g) ACK_N or ACK_1 frames with an ACK_CNT of 1 shall be transmitted during X_ID interlock (see 25.3.1 and 24.5.4), and in response to a connect-request (see 28.4.1).

h) If a Sequence Recipient receives a Data frame in Class 2 which falls within the SEQ_CNT range of a Recovery_Qualifier, it shall discard the Data Field of the frame and shall not deliver the Payload. The Sequence Recipient may transmit an ACK for the corresponding Data frame.

i) If a Sequence Initiator receives an ACK for a Data frame in Class 2 which falls within the SEQ_CNT range of a Recovery_Qualifier, it shall discard and ignore the ACK frame.

j) See 26.4.3.2, 26.4.3.3, and 26.4.3.1 for the role of acknowledgement frames (ACK) in flow control.

k) Each ACK shall be transmitted within an E_D_TOV timeout period of the event which prompts the initiative to transmit an ACK frame. For example, when using ACK_1, it shall be transmitted within E_D_TOV of the Data frame reception. Whereas, when using ACK_0, it shall be transmitted within E_D_TOV of receiving the last Data frame of the Sequence.

24.3.8 Normal Sequence completion

a) The Last Data frame of a Sequence shall be indicated by setting the F_CTL End_Sequence bit (F_CTL Bit 19) to one.

b) An Exchange Event shall be defined when the End_Sequence bit (Bit 19) is set = 1, and any of the following F_CTL bits are set as indicated:

- Sequence Initiative (Bit 16) = 1,
- Continue Sequence Condition (Bits 7-6) = 1 1,
- Invalidate X_ID (Bit 14) = 1,
- Last Sequence (Bit 20) = 1,
- Chained Sequence (Bit 17) = 1, or
- End_Connection (Bit 18) = 1.

c) A Sequence Event shall be defined when the End_Sequence bit (Bit 19) is set = 1, and the Continue Sequence Condition bits (Bits 7-6) in F_CTL are set to 0 0, 0 1, or 1 0 in the absence of an Exchange Event.

d) In Class 1 the Sequence Initiator shall consider a Sequence as deliverable and complete when it receives the final ACK for the Sequence (ACK with **EOFt** or **EOFdt** delimiter).

e) In Class 2 the Sequence Initiator shall consider a Sequence as deliverable (to the ULP) and complete when it receives the final ACK for the Sequence (ACK with **EOFt** delimiter). However, the Sequence Initiator shall account for all ACKs before reusing the SEQ_ID for this Exchange.

f) In Class 3 a Sequence Initiator shall consider a Sequence as deliverable and complete only if it has received a Basic Accept in reply to an ABTS frame or if it has received an Accept to a Read Exchange Status request which confirms the deliverability.

g) A Class 1 or 2 Sequence shall be considered complete by the Sequence Recipient if

- all Data frames are received,
- no Sequence errors are detected, and
- acknowledgements, if any, prior to acknowledgment of the last Data frame received have been transmitted.

h) A Class 3 Sequence shall be considered complete by the Sequence Recipient if

- all Data frames are received,
- no Sequence errors are detected, and
- the last Data frame is terminated by an **EOFt**.

i) For a normally completed Sequence in Class 1, the Sequence Recipient shall transmit an ACK frame (ACK_1, ACK_N, or ACK_0) with **EOFt** (or **EOFdt**) in response to the last Data frame of the Sequence (End_Sequence bit in F_CTL = 1) when the Sequence is deliverable. The End_Sequence bit in F_CTL of the ACK shall be set to one.

- In Discard multiple Sequences Error Policy, the Sequence is deliverable when all preceding ACK frames have been transmitted and the previous Sequence, if any, is deliverable.
- In Discard a single Sequence Error Policy, the Sequence is deliverable when all preceding ACK frames have been transmitted without regard to a previous Sequence.

j) In Class 2, if the last Data frame (End_Sequence = 1) transmitted is the last Data frame received for the Sequence, or if the last Data frame (End_Sequence = 1) received indicates an Exchange event (item b), the Sequence Recipient shall operate in the same manner as in Class 1 as specified in item i.

k) In Class 2, if the last Data frame (End_Sequence = 1) transmitted is not the last Data frame received and the last Data frame (End_Sequence = 1) transmitted indicates a Sequence Event, the Sequence Recipient may either:

- withhold transmission of the ACK corresponding to the last Data frame transmitted until all previous ACKs have been transmitted and the Sequence is deliverable, or
- save the value of End_Sequence (Bit 19) and Continue Sequence Condition (Bits 7-6), and transmit the ACK corresponding to the last Data frame with **EOFn** with End_Sequence and Continue Sequence bits set to zeros. When the last missing Data frame of the Sequence is received and the Sequence is deliverable, the Sequence Recipient shall transmit an ACK with **EOFt** with the value of End_Sequence and Continue Sequence Condition bits saved from the Data frame containing the End_Sequence bit = 1 with the SEQ_CNT and other fields which match the corresponding Data frame.

NOTE - When Sequences are being streamed in Class 2 with out of order delivery, transmission of ACK (**EOFn**) in response to the last Data frame of the Sequence (End_Sequence = 1) avoids costing the Initiator an extra Credit of one for the last Data frame of the Sequence while the Sequence Recipient waits for the last frame to be delivered.

l) In Class 1 or 2 the Sequence Initiator shall transmit the last Data frame with an **EOFn**.

m) In Class 3 the Sequence Initiator shall transmit the last Data frame with an **EOFt**.

n) In the last Data fram of a Sequence with the Sequenc Initiative held, the Sequence Initi-

ator may set the Continue Sequence bits in F_CTL to indicate that the next Sequence will be transmitted immediately (0 1), soon (1 0), or delayed (1 1). If Continue Sequence bits are set to 1 1, then the Sequence Initiator shall wait until the final ACK (**EOFt**) is received before initiating a new Sequence for this Exchange.

o) In the last Data frame of a Sequence, the Sequence Initiator shall set the

  – Sequence Initiative bit in F_CTL to 0 to hold Sequence Initiative.
  – Sequence Initiative bit in F_CTL to 1 to transfer Sequence Initiative.

  In Class 1 and 2, the Sequence Initiative is considered to be passed to the Sequence Recipient when the Sequence Initiator receives the final ACK (**EOFt**) of the Sequence with the Sequence Initiative bit = 1.

p) In the last Data frame of a Class 1 Sequence, the Sequence Initiator may indicate its desire to end the Dedicated Connection by setting the End_Connection (bit 18) = 1 in order to begin the procedure to remove the Dedicated Connection (see 28.7.2).

q) In the last Data frame of a Class 1 Sequence, the Sequence Initiator may indicate that it requires a reply Sequence be transmitted within the existing Dedicated Connection by setting the Sequence Initiative bit to one and Chained_Sequence bit to one in F_CTL.

r) Additional information regarding X_ID reassignment may also occur on the last Data frame of a Sequence (see 25.3.1).

s) After a Sequence is complete, the Sequence Recipient shall supply the status of the Sequence in the format of the Exchange Status Block defined in 24.8.1 in response to a Read Exchange Status Link Service command. Sequence status in the Exchange Status Block is available until X+2 Sequences have been completed (where X is the number of Open Sequences per Exchange supported by the Sequence Recipient as specified during Login) or the Exchange is terminated.

24.3.9 Detection f missing fram s

The following methods of detecting missing frames apply to a non-streamed Sequence or multiple streamed Sequences with continuously increasing SEQ_CNT.

a) In Class 1 a missing Data frame error is detected if a frame is received with a SEQ_CNT that is not one greater than the previously received frame, except when a SEQ_CNT wrap to zero occurs (e.g., SEQ_CNT = 3 received, then SEQ_CNT = 5 received; a missing frame error (SEQ_CNT = 4) is detected).

b) In Class 1 a missing Data frame error due to timeout is detected by the Sequence Recipient if a partial Sequence has been received and the next expected Data frame (current SEQ_CNT+1, except when a wrap to zero occurs) is not received within an E_D_TOV timeout period.

c) In Class 2 and 3, with out of order delivery, a potentially missing Data frame is detected if a frame is received with a SEQ_CNT that is not one greater than the previously received frame, except when a SEQ_CNT wrap to zero occurs. If the potentially missing Data frame is not received within the E_D_TOV timeout period, a missing frame error is detected.

d) In Class 2, with in order delivery, a potentially missing Data frame is detected if a frame is received with a SEQ_CNT that is not one greater than the previously received frame, except when a SEQ_CNT wrap to zero occurs. If the potentially missing Data frame is not received within the E_D_TOV timeout period, a missing frame error is detected.

NOTE - With in order delivery, a Class 2 frame may be delivered with its SEQ_CNT that is not one greater than the previously received frame, if a Class 2 frame which was transmitted earlier has been issued F_BSY or F_RJT. This frame is potentially missing, since it may be retransmitted.

e) In Class 3, with in order delivery, a missing Data frame is detected if a frame is received with a SEQ_CNT that is not one greater than the previously received frame, except when a SEQ_CNT wrap to zero occurs.

f) A Sequenc Recipient may also detect missing Class 2 or 3 Data frames through the use of a missing frame window. The size of the missing frame window, W, is set by the

Sequence Recipient and is not specified by FC-PH. A frame is considered missing by a Sequence Recipient if its SEQ_CNT is less than the highest SEQ_CNT received for that Sequence minus W. It is suggested that W be at least equal to End-to-end Credit.

NOTE - Fabric characteristics should be taken into account when attempting to establish a missing frame window - W. Too small a value may give false errors, whereas too large a value may create out of Credit conditions.

24.3.10 Sequence errors - Class 1 and 2

Errors within a Sequence may be detected by either the Sequence Initiator or the Sequence Recipient.

24.3.10.1 Rules common to all Discard policies

In discard policy, the Recipient shall discard the Data Field portion of Data frames (FC-2 Header is processed) received after the point at which the error is detected and including the frame in which the error was detected. In all cases except the Stop Sequence condition, the Sequence Recipient shall discard the entire Sequence. The following rules apply.

a) The types of Sequence errors that shall be detected by an N_Port include:

   - detection of a missing frame based on SEQ_CNT,
   - detection of a missing frame based on a timeout (E_D_TOV),
   - detection of an error within a frame (P_RJT),
   - reception of a Reject frame (F_RJT or P_RJT) or
   - detection of an internal malfunction.

b) If a Recipient receives a Data frame for a Sequence which the Recipient ULP wishes to stop receiving, the Recipient shall indicate the Stop Sequence condition to the Initiator by using the Abort Sequence Condition bits (1 0) in F_CTL. See 29.7.2.

c) If a Recipient detects an error within a valid frame of a Sequence, it shall indicate that error to the Initiator by transmitting a P_RJT with a reason code.

d) If a Recipient receives a Data frame for an Active Sequence which has previously had one or more Data frames rejected, the Recipient shall indicate that previous error to the Initiator on subsequent ACK frames using the Abort Sequence Condition bits (0 1) in F_CTL in the same manner as it would if a missing frame were detected.

e) If the Recipient has transmitted an ACK with the Abort Sequence Condition bits set, or a P_RJT in response to a Data frame, it shall post that information in the Sequence Status.

f) If an Initiator receives an ACK with the Abort Sequence Condition bits in F_CTL requesting Stop Sequence (1 0), it shall end the Sequence by transmitting the End_Sequence bit set to 1 in the next Data frame. If the last Data frame has already been transmitted, the Sequence Initiator shall not respond to the Stop Sequence request but shall notify the FC-4.

g) If an Initiator detects a missing frame, internal error, or receives an ACK with a detected rejectable condition, it shall abort the Sequence by transmitting an Abort Sequence (ABTS) Basic Link Service command (see 21.2.2).

h) If an Initiator receives an ACK with the Abort Sequence Condition bits (0 1) in F_CTL requesting that the Sequence be aborted, it shall abort the Sequence by transmitting an Abort Sequence (ABTS) Basic Link Service command (see 21.2.2).

i) If an Initiator receives a Reject frame (F_RJT, or P_RJT), it shall abort the Sequence by transmitting an Abort Sequence (ABTS) Basic Link Service command (see 21.2.2) if the Sequence has not been terminated by the Sequence Recipient or Fabric using an **EOFt** (**EOFdt**) on the RJT.

j) In Class 1, if the Sequence Initiator detects a Sequence timeout (see 29.2.4), it shall:

   - abort the Sequence using ABTS if end-to-end Credit is available,
   - perform the Link Reset Protocol if all Active Sequences have timed out or no end-to-end Credit is available, or
   - read status from the Recipient if the current state of a Sequence is uncertain in order to determine the existence of an error condition.

k) In Class 2, if the Sequence Initiator detects a Sequence timeout (see 29.2.4), it shall:

— transmit Link Credit Reset to the Recipient if no end-to-end Credit is available,

— abort the Sequence using ABTS when end-to-end Credit is available, or

— read status from the Recipient if the current state of a Sequence is uncertain in order to determine the existence of an error condition.

24.3.10.2 Discard multiple Sequences Error Policy

These rules apply to Discard multiple Sequences Error Policy and Discard multiple Sequences with Retransmission Error Policy.

a) In Class 1, if a Sequence Recipient detects a missing frame error, or detects an internal malfunction for a Sequence within an Exchange which requested Discard multiple Sequences Error Policy, it shall request that the Sequence be aborted by setting the Abort Sequence Condition bits (0 1) in F_CTL on the ACK corresponding to the Data frame during which the missing frame error was detected. For errors detected other than missing frame, the Abort Sequence Condition bits (0 1) in F_CTL shall be transmitted for any subsequent ACKs transmitted. The Sequence Recipient may continue to transmit ACKs for subsequent frames of the Sequence and any subsequent streamed Sequences until the ABTS frame is received. If an ACK is transmitted for the last Data frame of a Sequence, F_CTL bits 19, 18, 17, 16, and 14 settings on the Data frame shall be ignored and those bits shall be set to zero in the ACK frame in addition to bits 5-4 (0 1). See 29.7.1.

b) In Class 1, if a Sequence Recipient detects a missing frame error, or detects an internal malfunction for a Sequence within an Exchange which requested Discard multiple Sequences with Retransmission Error Policy, it requests that the Sequence be aborted and immediately retransmitted by setting the Abort Sequence Condition bits (1 1) in F_CTL on the ACK corresponding to the Data frame during which the missing frame error was detected.

If the Sequence Recipient is unable to transmit an ACK with the same SEQ_ID as the Sequence which requires retransmission, the Sequence Recipient shall follow the rules for Discard multiple Sequences Error Policy (bits 5-4 set to 0 1).

For errors detected other than missing frame, the Abort Sequence Condition bits (1 1) in F_CTL shall be transmitted for any subsequent ACKs transmitted. The Sequence Recipient may continue to transmit ACKs for subsequent frames of the Sequence and any subsequent streamed Sequences until it receives a new Sequence (**SOFix**) with the F_CTL Retransmission bit set = 1, or an ABTS frame is received. If an ACK is transmitted for the last Data frame of a Sequence, F_CTL bits 19, 18, 17, 16, and 14 settings on the Data frame shall be ignored and those bits shall be set to zero in the ACK frame in addition to bits 5-4 (1 1).

If the Sequence Recipient is unable to support Discard multiple Sequences with Retransmission Error Policy, it shall follow the rules for Discard multiple Sequences Error Policy (bits 5-4 set to 0 1). If the Sequence Initiator is unable to determine the correct Sequence boundary to begin retransmission, it shall either transmit the ABTS frame or Read Exchange Status (RES). See 29.7.1.

c) In Class 2, if a Sequence Recipient detects a missing frame error, transmits a P_RJT, or detects an internal malfunction for a Sequence within an Exchange which requested Discard multiple Sequences Error Policy, it shall request that the Sequence be aborted by setting the Abort Sequence Condition bits (0 1) in F_CTL on the ACK corresponding to the Data frame during which the missing frame error was detected. For detected errors other than missing frame, the Abort Sequence Condition bits (0 1) in F_CTL shall be transmitted for any subsequent ACKs transmitted. The Sequence Recipient may continue to transmit ACKs for subsequent frames of the Sequence and any subsequent streamed Sequences until the ABTS frame is received. Any ACKs transmitted for frames in this Sequence or any subsequent Sequences shall continue to set the Abort Sequence Condition bits to (0 1) (see 29.7.1). The last ACK for the Sequence in error shall be transmitted as described under normal Sequence completion.

d) In Class 1, if a Sequence Initiator receives an ACK with the Abort Sequence Condition bits (1 1) in F_CTL requesting that the Sequence be retransmitted, it shall begin retransmission of the first non-deliverable Sequence by starting a new Sequence and setting the

F_CTL Retransmission bit (F_CTL bit 9) set to 1 until it has received at least one ACK which indicates that the retransmitted Sequence has been successfully initiated by the Recipient. If the Sequence Initiator is unable to determine the correct Sequence boundary to begin retransmission, it shall either transmit the ABTS frame to abort the Sequence or Read Exchange Status using the RES Extended Link Services request. See 29.7.1.2.

24.3.10.3 Discard a single Sequence Error Policy

a) In Class 1 and 2, if a Sequence Recipient detects a missing frame error, or detects an internal malfunction for a Sequence within an Exchange which requested Discard a single Sequence Error Policy, it shall request that the Sequence be aborted by setting the Abort Sequence Condition bits (0 1) in F_CTL on the ACK corresponding to the Data frame during which the missing frame error was detected. For errors detected other than missing frame, the Abort Sequence Condition bits (0 1) in F_CTL shall be transmitted for any subsequent ACKs transmitted for this Sequence.

   The Sequence Recipient may continue to transmit ACKs for subsequent frames of the Sequence until the ABTS frame is received. However, it shall not continue to set the Abort Sequence Condition bits in any subsequent streamed Sequences. If the final ACK (**EOFt**) to the Sequence is transmitted, F_CTL bits 19, 18, 17, 16, and 14 settings on the Data frame shall be ignored and those bits shall be set to zero in the ACK frame in addition to bits 5-4 (0 1) (see 29.7.1).

24.3.10.4 Process with infinite buffers Error Policy

In process policy, the Recipient shall ignore errors detected on intermediate frames, or timeout errors such that ABTS is not requested. However, such errors shall be reported to an upper level.

a) If a Recipient detects an internal error related to a Sequence, or it detects that the first or last frame of a Sequence is missing, it shall request that the Sequence be aborted by setting the Abort Sequence Condition bits (0 1) in F_CTL on subsequent ACK frames. The Recipient shall continue to respond in the same manner as defined und r Discard a single Sequence Error Policy.

   NOTE - Missing last Data frame is detected by the Sequence timeout.

b) If a Sequence Recipient detects an error within a valid frame of a Sequence, it shall indicate that error to the Initiator by transmitting a P_RJT with a reason code.

24.3.11 Sequence errors - Class 3

Errors within a Sequence may only be detected by the Sequence Recipient. In both Discard policies, the Sequence Recipient shall discard Sequences in the same manner as in Class 1 and 2 with the exception that an ACK indication of Abort Sequence shall not be transmitted. In discard policy, the Recipient shall discard frames received after the point at which the error is detected. Individual FC-4s or upper levels may recover the entire Sequence or only that portion after which the error is detected.

a) The types of errors that shall be detected by an N_Port include:

   — detection of a missing frame based on timeout, or
   — detection of an internal malfunction.

b) If a Recipient detects an internal error, it shall abnormally terminate the Sequence, post the appropriate status, and notify the FC-4 or upper level. One or more Sequences shall not be delivered based on single or multiple Sequence discard Error Policy.

c) If a Recipient detects a missing frame, it shall abnormally terminate the Sequence, post the appropriate status, and notify the FC-4 or upper level. One or more Sequences shall not be delivered based on single or multiple Sequence discard Error Policy.

d) In the Discard multiple Sequences Error Policy in Class 3, the Sequence Recipient shall not be required to utilize a timeout value of R_A_TOV following detection of a missing frame. Therefore, frames may be discarded for an Exchange forever if other detection mechanisms ar not utilized by the Sequence Initiator.

e) Notification of the Sequ nce error condition to the Initiator is the responsibility of th FC-4 or upper level.

24.3.12 Sequ nce Status Rules

The following rules summarize handling of Sequence Status Block processing:

a) A Sequence shall be considered Open and Active by the Sequence Initiator after transmission of the first frame of the Sequence. The Sequence shall remain Active until the Sequence Initiator has transmitted the last frame of the Sequence. The Sequence Initiator shall consider the Sequence Open until

- it receives the ACK (**EOFt**),
- Basic Accept is received to an ABTS frame,
- A Read Sequence Status or Read Exchange Status indicates normal or abnormal completion,
- the Link Reset Primitive Sequence is received or transmitted during a Dedicated Connection,
- the Exchange is aborted using the ABTX Link Service request, or
- a Logout Link Service request is completed.

b) A Sequence shall be considered Open and Active, and an SSB Opened, by the Sequence Recipient when any frame in a Sequence is received for the first Sequence of a new Exchange as indicated in F_CTL bit 21. An Exchange Status Block is Opened at the same time and the Exchange becomes Active.

c) A Sequence shall be considered Open and Active, and an SSB Opened, by the Sequence Recipient when any frame in a Sequence is received for an Open Exchange.

d) If the Sequence Recipient transmits an ACK frame with the Abort Sequence Condition bits other than 0 0, it shall post that status in the Sequence Status Block status.

e) If a Sequence completes normally and is deliverable, its status shall be posted in the Sequence Status Block.

f) If a Sequence completes abnormally by the Abort Sequence Protocol, its status shall be posted in the Sequence Status Block.

g) The Exchange Status Block shall be updated with Sequence Status information when the Sequence becomes abnormally complete, or normally complete.

24.3.13 Exchange termination

a) The last Sequence of the Exchange shall be indicated by setting the F_CTL Last_Sequence bit = 1 in the last Data frame of a Sequence. The Last_Sequence bit may be set to 1 prior to the last Data frame; once it has been set = 1, it shall remain set for the remainder of the Sequence.

b) The Exchange shall be terminated when the last Sequence is completed by normal Sequence completion rules.

c) An Exchange may be abnormally terminated using the ABTX Extended Link Services request. A Recovery_Qualifier range timeout may be required in Class 2 and 3.

d) An Exchange shall be abnormally terminated following Logout with the other N_Port involved in the Exchange (either Originator or Responder). A Recovery_Qualifier range timeout may be required in Class 2 and 3.

24.3.14 Exchange Status Rules

The following rules summarize handling of Exchange Status Block processing:

a) An Exchange shall be considered Active, and an ESB Opened, by the Originator after transmission of the first frame of the first Sequence.

b) An Exchange shall be considered Active, and an ESB Opened, by the Sequence Recipient when any frame in the first Sequence is received.

c) An Exchange shall be remain Open until

- the last Sequence of the Exchange completes normally,
- a timeout period of E_D_TOV has elapsed since the last Sequence of the Exchange completed abnormally, or
- the Exchange is successfully aborted with ABTX (which includes a Recovery_Qualifier timeout, if necessary).

d) When an Exchange is no longer Open, it shall be complete and the Exchange resources associated with the Exchange, including the Exchange Status Block, are available for reuse. An upper level may choose to complete an Exchange with an interlocked protocol in order to ensure that both the Originator and Responder agree that the

Exchange is complete. Such a protocol is outside the scope of FC-PH.

e) The contents of an Exchange Status Block may be made available (RES request) until the resource is allocated to another Exchange.

24.4 Exchange management

An Exchange is managed as a series of unidirectional Data frame Sequences. The initial Sequence shall be transmitted by the Originator of the Exchange. Control and intermixing of Sequences within an Exchange are identified by F_CTL bits within the Frame Header.

Following the initial Sequence, subsequent Sequences may be transmitted by either the Originator or the Responder facilities based on which facility holds the Sequence Initiative.

24.5 Exchange origination

The key facilities, functions, and events involved in the origination of an Exchange by both the Originator and Responder are diagrammed in figure 70. An Exchange for Data transfer may be originated with a destination N_Port following destination N_Port Login. Login provides information necessary for managing an Exchange and Sequences such as: Class, the number of Concurrent Sequences, Credit, and Receive Data Field Size. Login and Service Parameters are discussed in 23.6. An Exchange is originated through the initiation of a Sequence. The rules in 24.3.2 specify the requirements for originating an Exchange.

24.5.1 Exchange Originator

When an Exchange is originated by an N_Port, that N_Port shall assign an Originator Exchange_ID (OX_ID) unique to that N_Port Identifier. An Originator Exchange Status Block associated with the OX_ID is allocated and bound to the Exchange and other link facilities in that N_Port for the duration of the Exchange. All frames associated with that Exchange contain the assigned OX_ID except during OX_ID reassignment (when OX_ID is hex 'FFFF'). The status of the Exchange shall be tracked by the Originator in the Originator Exchange Status Block. See 24.6.2 for more information on unique Sequence_Qualifiers.

Each frame within the Exchange transmitted by the Originator shall be identified with an F_CTL bit designating the frame as Originator generated (Exchange Context). The Originator Exchange_ID provides the mechanism for tracking Sequences for multiple concurrent Exchanges which may be Active at the same time.

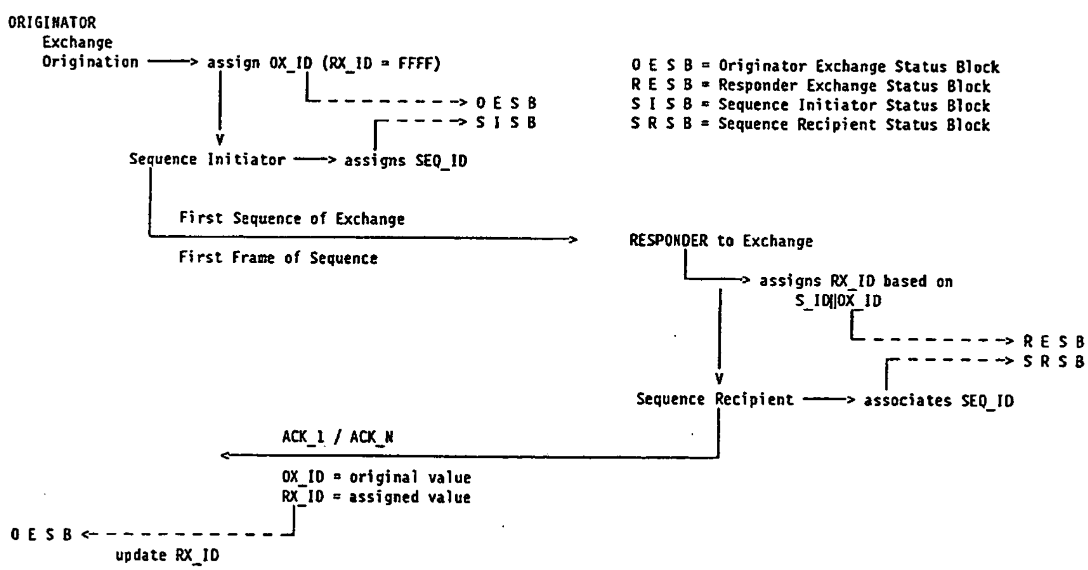


Figure 70 - Exchange origination

NOTE - Since the Originator assigns the OX_ID, assignment may be organized to provide efficient processing within the N_Port.

24.5.2 Exchange Responder

The destination N_Port shall be designated as the Responder for the duration of the Exchange. When the destination N_Port receives the first Sequence of the Exchange, that N_Port shall assign a Responder Exchange_ID (RX_ID) to the newly established Exchange. This RX_ID is associated with the OX_ID from a given S_ID to a Responder Exchange Status Block (S_ID||OX_ID). See 24.6.2 for more information on unique Sequence_Qualifiers.

In Class 1 and 2, the assigned RX_ID shall be transmitted to the Originator on the ACK frame responding to the last Data frame of the Sequence or earlier, if possible. In a Class 3 bidirectional Exchange, the assigned RX_ID shall be transmitted to the Originator in the first Data frame transmitted by the Responder. If X_ID interlock has been specified during Login by the Sequence Recipient, the RX_ID shall be assigned in the ACK to the first Data frame of the Sequence. The Originator shall withhold additional frame transmission for the Exchange until the ACK is received. The Responder Exchange_ID provides the mechanism for tracking Sequences for multiple concurrent Exchanges from multiple S_IDs or the same S_ID.

NOTE - Since the Responder assigns the RX_ID, assignment may be organized to provide efficient processing within the N_Port.

Each frame within the Exchange transmitted by the Responder is identified with an F_CTL bit designating the frame as Responder generated (Exchange Context). Each frame within the Exchange transmitted by the Responder is identified with the assigned RX_ID except during RX_ID reassignment (when the RX_ID is hex 'FFFF'). The status of the Exchange shall be tracked by the Responder in the Responder Exchange Status Block.

24.5.3 X_ID assignment

In the first frame of an Exchange, the Originator shall set the OX_ID to an assigned value and the RX_ID value to hex 'FFFF' (unassigned). When the Responder receives the first Sequence of an Exchange, it shall assign an RX_ID and shall return the RX_ID in the ACK frame sent in response to the last Data frame in the Sequence, or in an earlier ACK (Class 1 and 2). In a Class 3 bidirectional Exchange, the Responder shall assign an RX_ID in the first Data frame transmitted.

If the Responder N_Port has indicated during Login that an X_ID interlock is required at X_ID assignment and reassignment transitions, then the Originator shall not transmit subsequent frames of the Exchange until the corresponding ACK has been received with an assigned X_ID for the Exchange Responder. The Originator shall set the RX_ID to Hex 'FFFF' until the assigned RX_ID is received. When the Originator receives the assigned RX_ID, it shall set the RX_ID field to the assigned value for all subsequent frames.

For all remaining frames within the Exchange, OX_ID and RX_ID fields retain these assigned values unless either the Originator or the Responder invalidates its X_ID value at the end of a Sequence (see 24.5.4, 25.3.1, and 25.3.2 for X_ID reassignment).

24.5.4 X_ID interlock

X_ID interlock is only applicable to Classes 1 and 2. When an N_Port initiates a Sequence with an N_Port which has specified during Login that X_ID interlock is required and the Recipient's X_ID is invalid or unassigned, the initiating N_Port shall transmit the first frame of the Sequence with the Recipient's X_ID set to hex 'FFFF' and shall withhold transmission of additional frames until the corresponding ACK with an assigned X_ID has been received from the Recipient. The assigned X_ID is then used in all subsequent frames in the Sequence.

24.6 Sequenc management

24.6.1 Active and Open Sequence

From the standpoint of the Sequence Initiator, a Sequence is Active for the period of time from the transmission of the Data frame initiating the Sequence until the end of the last Data frame of the Sequence is transmitted. In Class 1, the Sequence Initiator considers the Sequence Open until the ACK with **EOFt** (**EOFdt**) is received, the Sequence is aborted by performing the ABTS Protocol, or the Sequence is abnormally terminated. In Class 2, the Sequence Initiator considers the Sequence Open until the ACK with **EOFt** is received, the Sequence is aborted by performing the ABTS Protocol, or the Sequence is abnormally terminated. In Class 3, the Sequence Initiator considers the Sequence Active until the **EOFt** is transmitted. In Class 3, the Sequence Initiator considers the Sequence Open until the deliverability is confirmed or an R_A_TOV timeout period has expired.

In Class 1 and 2, from the standpoint of the Sequence Recipient, a Sequence is Active and Open from the time any Data frame is received until the **EOFt** (**EOFdt**) is transmitted in the ACK to the last Data frame, or abnormal termination of the Sequence. In Class 3, from the standpoint of the Sequence Recipient, a Sequence is Active and Open from the time the initiating Data frame is received until all Data frames up to the frame containing **EOFt** have been received.

See 24.6.2 for more information regarding reuse of SEQ_ID and Sequence_Qualifiers.

24.6.2 Sequence_Qualifier management

The Sequence Initiator assigns a SEQ_ID which is unique between the N_Port pair of the Initiator and the Recipient at the time it is assigned and until the Sequence is complete. When the Sequence completes normally or abnormally, the SEQ_ID is reusable by the Sequence Initiator for any Sequence_Qualifier, including the same Recipient and Exchange providing that Sequence rules are followed (see 24.3.4). If a Sequence is aborted using the Abort Sequence Protocol or the ABTX Link S rvice request, a Recovery_Qualifier range may be specified by the Sequence Recipi nt (see 29.7.1.1), however, SEQ_ID shall not be included in the Recovery_Qualifier.

24.6.3 Sequence initiative and termination

When a Sequence is terminated in a normal manner, the last Data frame transmitted by the Sequence Initiator is used to identify two conditions:

a) Sequence initiative, and
b) Sequence termination.

24.6.4 Transfer of Sequence Initiative

The Sequence Initiator controls which N_Port shall be allowed to initiate the next Sequence for the Exchange. The Sequence Initiator may hold the initiative to transmit the next Sequence of the Exchange or the Sequence Initiator may transfer the initiative to transmit the next Sequence of the Exchange. The decision to hold or transfer initiative shall be indicated by Sequence_Initiative bit in F_CTL.

In Class 1 and 2, the Sequence Recipient shall not consider Sequence Initiative to have been passed until the Sequence which passes the Sequence Initiative is completed successfully and the ACK (**EOFt**) has been transmitted with the Sequence Initiative bit (F_CTL bit 16) = 1.

In Class 1 and 2, when a Sequence Initiator detects a Data frame from the Recipient for an Exchange in which it holds the Sequence Initiative, it shall transmit a P_RJT with a reason code of Exchange error (excluding the ABTS frame). In Class 3, when a Sequence Initiator detects a Data frame (excluding the ABTS frame) from the Recipient for an Exchange in which it holds the Sequence Initiative, it shall abnormally terminate the Exchange and discard all frames for the Exchange.

End_Sequence

When the Sequence Initiator is ending the current Sequence, it shall set the End_Sequence bit in F_CTL to one on the last Data frame of the Sequence.

24.6.5 Sequ nce termination

The last Data frame transmitted by the Sequence Initiator is indicated by setting the End_Sequence bit in F_CTL to one. The Sequence is terminated by either the Initiator or the Recipient transmitting a frame terminated by **EOFt**. The Sequence Initiator is in control of terminating the Sequence. Transmission of the **EOFt** may occur in two ways:

- In Class 1 and 2, the Sequence Recipient transmits an ACK frame of ACK_1, ACK_N, or ACK_0 with **EOFt** in response to the last Data frame received for the Sequence.
- In Class 3, the Sequence Initiator transmits the last Data frame of the Sequence with **EOFt**.

Class 1

When **EOFt** has been transmitted or received by each N_Port, the appropriate Exchange Status Block associated with the Sequence shall be updated in each N_Port to indicate that the Sequence was completed and whether the Originator or Responder facility holds the Sequence Initiative. Link facilities associated with the Sequence (including Sequence Status Block) are released and available for other use.

Class 2

Since Class 2 frames may be delivered out of order, Sequence processing is only completed after all frames (both Data and ACK) have been received, accounted for, and processed by the respective N_Ports.

When the Sequence is completed by each N_Port, the appropriate Exchange Status Block associated with the Sequence shall be updated in each N_Port to indicate that the Sequence was completed and whether the Originator or Responder facility holds the Sequence Initiative. Link facilities associated with the Sequence (including the Sequence Status Block) are released and available for other use.

NOTE - Since ACKs may arrive out of order, the Sequence Initiator may receive the ACK which contains **EOFt** before ACKs for the same Sequence. The Sequence Initiator shall not consider the Sequence normally terminated until it has received the final ACK (see 29.7.3).

Class 3

The Sequence Initiator shall terminate the last Data frame of the Sequence with **EOFt**. Acknowledgment of Sequence completion is the responsibility of the Upper Level Protocol.

When the Sequence is completed by each N_Port, the appropriate Exchange Status Block associated with the Sequence shall be updated in each N_Port to indicate that the Sequence was completed and whether the Originator or Responder facility holds the Sequence Initiative. Link facilities associated with the Sequence (including Sequence Status Block) are released and available for other use.

Continue Sequence Condition

When an N_Port terminates a Sequence yet holds the Sequence Initiative, the N_Port may include additional information regarding the timing of the next Sequence. The N_Port uses the Continue Sequence Condition bits in F_CTL to indicate that the next Sequence is being transmitted immediately (0 1), soon (1 0), or delayed (1 1). The estimate of time is based on the time to remove and reestablish a Class 1 Connection, regardless of which Class of Service is being used. This allows the destination N_Port to Invalidate its X_ID if it chooses to reallocate resources for a delayed Sequence.

Chained_Sequence (C_S)

Certain existing system architectures require immediate feedback during specific phases of an I/O operation. The Chained_Sequence (C_S) bit in F_CTL may be set to one on the last Data frame of a Sequence when the Sequence Initiative is being transferred to indicate that a reply Sequence is required from the Sequence Recipient before a Dedicated Connection is removed. The presence of the C_S bit, End_Sequence bit, and transfer of Sequence Initiative overrides the E_C bit if previously set by the Sequence Recipient (see 28.7.1 and 28.7.2).

End_Sequence

When the S quence Initiator is ending the current Sequence, it shall set the End_Sequence bit in F_CTL to one on the last Data frame of the Sequence.

24.7 Exchange termination

24.7.1 Normal termination

An Exchange may be terminated by either the Originator or the Responder. The facility terminating the Exchange shall indicate Exchange termination on the the last Sequence of the Exchange by setting the Last_Sequence bit in F_CTL on the last frame, or earlier, if possible, of the last Sequence of the Exchange.

The Sequence shall be terminated according to normal Sequence termination rules. When the last Sequence of the Exchange is terminated normally, the Exchange shall also be terminated. The OX_ID and RX_ID and associated Exchange Status Blocks are released and available for reuse.

24.7.2 Abnormal termination

An Exchange may be abnormally terminated by either the Originator or the Responder by using the Abort Exchange Protocol or Sequence timeout of the last Sequence of the Exchange. In general, reception of a reject frame with an action code of 2 is not recoverable at the Sequence level and aborting of the Exchange is probable. Other reasons to abort an Exchange are FC-4 protocol dependent and not defined within FC-PH.

24.8 Status blocks

24.8.1 Exchange Status Block

An Exchange Status Block is a logical construct used to associate the OX_ID, RX_ID, D_ID and S_ID of an Exchange. The Status Block is used throughout the Exchange to track the progress of the Exchange and identify which N_Port holds the initiative to transmit Sequences. The Exchange Status Block shall continue to exist even following X_ID Invalidation, since the Operation_Associator is used to locate the ESB (use of a Process_Associator may also be required).

The Exchange Status Block shall record Exchange status information and Sequence Status for a number of Sequences received as a Sequence Recipient which is supplied in the RES Link Services request. Equivalent information to track transmitted Sequences is required by the Sequence Initiator for internal tracking of Exchange progress but is not required to be supplied to any other N_Port. The Sequence Status is stored in the Exchange Status Block in the oldest to newest order. The oldest Sequence is dropped out of the ESB when new Sequence status is added.

Table 103 - Exchange Status Block

Item	Size -Bytes
OX_ID	2
RX_ID	2
Originator Address Identifier (High order byte - reserved)	4
Responder Address Identifier (High order byte - reserved)	4
E_STAT	4
reserved	4
Service Parameters	112
Oldest Sequence Status (SSB format-first 8 bytes)	8
Intermediate Sequence Status (SSB format-first 8 bytes)	X × 8
Newest Sequence Status (SSB format-first 8 bytes)	8

E_STAT

Bit 31 - ESB owner

0 = Originator
1 = Responder

Bit 30 - Sequence Initiative

0 = Other Port holds initiative
1 = This Port holds initiative

Bit 29 - Completion

0 = Open
1 = complete

Bit 28 - Ending Condition

0 = normal
1 = abnormal

Bit 27 - Error typ

0 = Exchange aborted with ABTX
1 = Exchange abnormally terminated

Error type is only valid when bit 28 is set to one.

Bit 26 - Recovery_Qualifier

0 = none
1 = Active

Bits 25-24 - Exchange Error Policy

0 0 = Abort, Discard multiple Sequences
0 1 = Abort, Discard a single Sequence
1 0 = Process with infinite buffers
1 1 = Discard multiple Sequences with immediate retransmission

Bit 23 - Originator X_ID invalid

0 = Originator X_ID valid
1 = Originator X_ID invalid

X_ID validity status reflects the completion condition of the newest Sequence Status Block contained in the ESB.

Bit 22 - Responder X_ID invalid

0 = Responder X_ID valid
1 = Responder X_ID invalid

X_ID validity status reflects the completion condition of the newest Sequence Status Block contained in the ESB.

Bits 21-0 - Reserved

24.8.2 Sequence Status Block

A Sequence Status Block is a logical construct used to track the progress of a single Sequence by an N_Port on a frame by frame basis. A Sequence Status Block shall be Opened and maintained by the Sequence Recipient for each Sequence received in order to support the RSS Link Service request. Information equivalent to an SSB is required for the Sequence Initiator to track Sequence progress internally, but is not required to be supplied to any other N_Port.

Table 104 - Sequence Status Block

Item	Size -Bytes
SEQ_ID	1
reserved	1
Lowest SEQ_CNT	2
Highest SEQ_CNT	2
S_STAT	2
Error SEQ_CNT	2
OX_ID	2
RX_ID	2
reserved	2

S_STAT

Bit 15 - Sequence context

0 = Initiator
1 = Recipient

Bit 14 - Open

0 = not Open
1 = Open

Bit 13 - Active

0 = not Active
1 = Active

Bit 12 - Ending Condition

0 = normal
1 = abnormal

Bits 11 - 7 clarify the reason for an abnormal ending condition.

Bits 11 - 10 ACK, Abort Sequence condition

0 0 = continue
0 1 = Abort Sequence requested
1 0 = Stop Sequence requested
1 1 = Abort with Retransmission requested

Bit 9 - ABTS protocol performed

0 = ABTS not completed
1 = ABTS completed by Recipient

Bit 8 - Retransmission performed

0 = Retransmission not completed
1 = Retransmission completed by Recipient

Bit 7 - Sequenc time-out

0 = Sequenc not timed-out
1 = Sequence timed-out by R cipient

(E_D_TOV)

Bit 6 - P_RJT transmitted

0 = P_RJT not transmitted
1 = P_RJT transmitted

Bits 5-4 Class

0 0 = reserved
0 1 = Class 1
1 0 = Class 2
1 1 = Class 3

Bit 3 - ACK (EOFt) transmitted

0 = ACK (EOFt) not transmitted
1 = ACK (EOFt or EOFdt) transmitted

Bits 2-0 - Reserved

25 Association Header management and usage

25.1 Introduction

FC_PH provides the Exchange facility as the fundamental mechanism for coordinating the interchange of information and data between two N_Ports or nodes. All Data transmission shall be part of an Exchange. Certain system architectures may use an Association Header to provide two additional, but separate, functions.

A Process_Associator may be used to provide one or more of the following functions:

- provide a method to direct an Exchange to a single system image where multiple system images share an N_Port facility,
- provide a method to direct an Exchange to a single process where multiple processes share an N_Port facility,
- identify a source or destination process, or a source or destination system image which shall be retained beyond the life of an Exchange for certain FC-4s. In this instance, the Process_Associator in combination with the N_Port Identifier is used to identify a process or image related to a previous Exchange.

An Operation_Associator may be used to provide one or more of the following functions:

- provide a method to manage an I/O Operation consisting of more than one simultaneously active Exchange,
- provide a method to map or relate an Exchange to existing system control blocks or structures used by an I/O subsystem, or
- provide a method to limit the number of Exchange management facilities (such as X_IDs) in use at any given time.

There are two N_Port Login options which may be required, supported, or not supported by an N_Port in Class 1 and 2

- Initial Process_Associator, and
- X_ID reassignment.

Login only completes successfully (Accept) if each N_Port's requirements are satisfied. If an N_Port either supports or requires an Initial Process_Associator or X_ID r assignment, then it shall also require X_ID interlock (Class 1 and 2), as specified in Login. The following list specifies the need for an Association Header for a specific Exchange:

If neither N_Port requires an Initial Process_Associator or X_ID reassignment,
—no Association Header is required.

If either N_Port requires X_ID reassignment,
—an Association Header is required.

If both N_Ports require an Initial Process_Associator,
—an Association Header is required.

If only one N_Port requires an Initial Process_Associator and the other N_Port supports it,
—an Association Header may be required.

25.2 Establishing the final Association Header

The rules which specify the procedures for Association Header management are in addition to all the rules and behavior described in other clauses of FC_PH. The Association Header is composed of information from the Originator and from the Responder. The method by which this composite information is obtained is by transmission of a first Association Header by the Originator which is followed by transmission of the final Association Header by the Responder in a later Sequence to reflect one or both of the Responder Associators. The procedure is simplified by following the first and final Association Header rules. The procedure begins with the origination of the Exchange. The content of the Association Header is specified in 19.4.

25.2.1 Exchange Origination

If either N_Port requires X_ID reassignment or the Responder requires an Initial Process_Associator, the Originator of an Exchange shall transmit an Association Header in the first Data frame of the Exchange.

If neither N_Port requires X_ID reassignment and the Responder does not require an Initial Process_Associator, then an Association Header is not required for the Exchange.

Meaningful values (validity byte in Word 0 of Association Header) are implementation dependent and have no constraints placed on their values other than those defined in 19.4.

The Association Header to be transmitted by the Originator shall be composed as follows:

- the Originator Process_Associator shall be set to
  a) a meaningful value if the Originator requires an Initial Process_Associator, or
  b) validity indicated as not meaningful.
- the Originator Operation_Associator shall be set to
  a) a meaningful value if the Originator requires X_ID reassignment, or
  b) validity indicated as not meaningful.
- the Responder Process_Associator shall be set to
  a) a meaningful value if the Responder requires an Initial Process_Associator (obtained from a Name_Server or by other means outside the scope of FC_PH), or
  b) validity indicated as not meaningful.
- the Responder Operation_Associator shall be set to
  a) validity indicated as not meaningful.

25.2.2 First transfer rules

The following rules apply to the transmission of the first Association Header transmitted by the Originator:

a) the Originator shall assign a value to OX_ID and set RX_ID to hex 'FFFF' and include the Association Header on the first Data frame of the Exchange.
b) X_ID interlock is required on the first Data frame of the Exchange.
c) the RX_ID shall be specified in the ACK to the first Data frame of the Exchange.
d) the first Association Header transfer is confirmed by receiving the ACK to the first Data frame of the Exchange.
e) after the Originator has confirmed Association Header transfer, it may invalidate its X_ID if it requires X_ID reassignment (see 25.3.1) according to the rules for X_ID invalidation.
f) if the first Association Header transfer is not confirmed, the Originator shall transmit an ABTS frame to either Abort th Sequence or confirm transfer.
g) The Responder shall not invalidate its X_ID until it has confirmed that the final Association Header has been transferred (see 25.2.3).
h) If the Responder does not require an Operation_Associator, the first Association Header transmitted by the Originator shall be the final Association Header.

25.2.3 Responder Sequence Initiative

When the Exchange Responder receives the Sequence Initiative for the first time during an Exchange, if ever, which contained an Association Header in the first Data frame of the Exchange, the Exchange Responder shall transmit an Association Header on the first Data frame transmitted if the Responder requires a Responder Operation_Associator. The final Association Header shall be composed as follows:

- the Originator Process_Associator shall be set to
  a) the value received in the first Association Header.
- the Originator Operation_Associator shall be set to
  a) the value received in the first Association Header.
- the Responder Process_Associator shall be set to
  a) the value received in the first Association Header if the Responder requires an Initial Process_Associator, or
  b) validity indicated as not meaningful.
- the Responder Operation_Associator shall be set to
  a) a meaningful value if the Responder requires X_ID reassignment.

25.2.4 Final transfer rules

The Association Header transmitted by the Responder becomes the final Association Header for the Exchange. That is, whenever an Association Header is required to be transferred, the Association Header designated as final shall be transferred. The Association Header shall not change its content throughout the remainder of the Exchange.

The following rules apply to the Association Header transmitted by the Responder.

a) the Responder shall use the RX_ID value assigned when the Exchange was originated.

b) if the OX_ID is valid, the Responder shall use that value.

c) if the OX_ID was pr viously invalidated, the Responder shall set OX_ID = hex 'FFFF'. Since X_ID interlock is required, the Responder shall wait for the ACK to the first Data frame before transmitting any subsequent Data frames for the Exchange.

d) the final Association Header transfer is confirmed by receiving an ACK to the first Data frame of the first Sequence Initiative transmitted by the Responder, or by confirming that the first Data frame was successfully received.

e) after the Responder has confirmed final Association Header transfer, it may invalidate its X_ID if it requires X_ID reassignment (see 25.3.1) according to the rules for X_ID invalidation.

f) if the final Association Header transfer is not confirmed, the Responder shall transmit an ABTS frame to either Abort the Sequence or confirm transfer.

g) the Responder shall not invalidate its X_ID until it has confirmed that the final Association Header has been transferred.

h) both the Originator and Responder shall use the final Association Header whenever an Association Header is required to be transmitted for the duration of the Exchange.

25.3 Association Header Usage

After the first and final Association Headers have been transferred, the Association Header is used when the destination N_Port has invalidated its X_ID. The Association Header associated with an Exchange shall also be included in the Payload of certain Extended Link Service commands (see 25.3.3) when the commands reference that Exchange.

An example of a system using X_ID Invalidation is included in annex R.

25.3.1 X_ID Invalidation

X_ID invalidation shall only be performed in Class 1 and 2. An N_Port shall only invalidate its X_ID if it had previously indicated in Login that it required X_ID reassignment. An X_ID shall not be invalidated under conditions other than those specified in the following rules. The following rules specify the conditions under which an N_Port may invalidate its X_ID (OX_ID for Originator, RX_ID for Responder). Either the Originator or Responder may be the Sequence Initiator or Recipient in the following rules.

a) In the last Data frame of a Sequence, the Sequence Initiator shall set the

- Invalidate X_ID bit in F_CTL to 0 to retain X_ID assignment.
- Invalidate X_ID bit in F_CTL to 1 to invalidate its current X_ID.

b) In the last Data frame of a Sequence with the Sequence Initiative held, the Sequence Initiator may set the Continue Sequence bits in F_CTL to indicate that the next Sequence will be delayed (1 1). If Continue Sequence bits are set to 1 1, then the Sequence Recipient shall set the

- Invalidate X_ID bit in F_CTL to 0 to retain its X_ID assignment in the ACK response.
- Invalidate X_ID bit in F_CTL to 1 to invalidate its current X_ID in the ACK response.

c) If the last Data frame of a Sequence has the Invalidate X_ID bit set to one, the Sequence Recipient shall set the

- Invalidate X_ID bit in F_CTL to 0 to retain its X_ID assignment in the ACK response.
- Invalidate X_ID bit in F_CTL to 1 to invalidate its current X_ID in the ACK response.

d) If the last Data frame of a Sequence transfers Sequence Initiative, the Sequence Recipient shall set the

- Invalidate X_ID bit in F_CTL to 0 to retain its X_ID assignment in the ACK response.
- Invalidate X_ID bit in F_CTL to 1 to invalidate its current X_ID in the ACK response.

25.3.2 X_ID reassignment

When an N_Port initiates a Sequence, there are five cases to consider relative to X_ID reassignment:

- Case 1. Originator as Sequence Initiator, RX_ID previously invalidated, OX_ID valid or previously invalidated.

- Case 2. Responder as Sequence Initiator, OX_ID previously invalidated, RX_ID valid or previously invalidated.

- Cas 3. Originator as S quence Initiator, OX_ID previously invalidated, RX_ID valid.

— Case 4. Responder as Sequence Initiator, RX_ID previously invalidated, OX_ID valid.

— Case 5. Either Originator or Responder as Sequence Initiator, both OX_ID and RX_ID valid.

Case 1 and Case 2 require transmission of the Association Header. The other cases do not require transmission of the Association Header.

25.3.2.1 Case 1

If the Originator of an Exchange initiates a Sequence (Sequence Initiator) with the Responder (Sequence Recipient) and the Responder has previously invalidated its X_ID, the Originator shall

— transmit the final Association Header,
— transmit the RX_ID = FFFF (X_ID interlock required), and
— either set OX_ID to its current value, or
— set OX_ID to a new value if the Originator had invalidated its X_ID in the previous Sequence.

Reassignment rules

a) an Originator or Responder shall only reassign its X_ID value if its X_ID had been previously invalidated.

b) if the Originator reassigns its OX_ID to a new value, it shall set the X_ID reassigned bit (F_CTL bit 15) = 1 on the first Data frame of the Sequence.

c) the Responder shall assign an RX_ID value and set the X_ID reassigned bit = 1 in the ACK to the first Data frame of the Sequence.

d) the Originator shall update the RX_ID to its reassigned value in Data frames transmitted in the Sequence after the ACK frame is received which has the X_ID reassigned bit set = 1. The reassigned RX_ID shall be used until the Responder invalidates its X_ID, if ever.

e) both the Originator and Responder shall update the Exchange Status Block with a reassigned OX_ID, if any, and the reassigned RX_ID.

f) the X_ID invalidation status shall be updated to reflect the current X_ID status in the Exchange Status Block when the Sequence completes.

25.3.2.2 Case 2

If the Responder of an Exchange initiates a Sequence (Sequence Initiator) with the Originator (Sequence Recipient) and the Originator has previously invalidated its X_ID, the Responder shall

— transmit the final Association Header, if established, or
— transmit the first Association Header with updates to the Responder Associators as specified in 25.2.4,
— transmit the OX_ID = FFFF (X_ID interlock required), and
— either set RX_ID to its current value, or
— set RX_ID to a new value if the Responder had invalidated its X_ID in the previous Sequence.

Reassignment rules

a) an Originator or Responder shall only reassign its X_ID value if the X_ID had been previously invalidated.

b) if the Responder reassigns its RX_ID to a new value, it shall set the X_ID reassigned bit (F_CTL bit 15) = 1 on the first Data frame of the Sequence.

c) the Originator shall assign an OX_ID value and set the X_ID reassigned bit = 1 in the ACK to the first Data frame of the Sequence.

d) the Responder shall update the OX_ID to its reassigned value in Data frames transmitted in the Sequence after an ACK frame is received which has the X_ID reassigned bit set = 1. The reassigned OX_ID shall be used until the Originator invalidates its X_ID, if ever.

e) both the Originator and Responder shall update the Exchange Status Block with a reassigned RX_ID, if any, and the reassigned OX_ID.

f) the X_ID invalidation status shall be updated to reflect the current X_ID status in the Exchange Status Block when the Sequence completes.

25.3.2.3 Case 3

If the Originator initiates a Sequence and the RX_ID is valid, the Originator shall not transmit an Association Header. The Originator shall set its OX_ID to a new value and it shall set the X_ID reassigned bit (F_CTL bit 15) = 1 on the first Data frame of the Sequence and each subsequent Data frame until an ACK is received with the OX_ID = reassigned value. If it does not detect confirmation of the reassigned OX_ID, it shall perform the Abort Sequence Protocol. If the Abort Sequence Protocol is unsuccessful (BA_RJT), it shall read the Exchange Status Block.

25.3.2.4 Case 4

If the Responder initiates a Sequence and the OX_ID is valid, the Responder shall not transmit an Association Header. The Responder shall set its RX_ID to a new value and it shall set the X_ID reassigned bit (F_CTL bit 15) = 1 on the first Data frame of the Sequence and each subsequent Data frame until an ACK is received with the RX_ID = reassigned value. If it does not detect confirmation of the reassigned RX_ID, it shall perform the Abort Sequence Protocol. If the Abort Sequence Protocol is unsuccessful (BA_RJT), it shall read the Exchange Status Block.

25.3.2.5 Case 5

If both X_IDs are valid, the Association Header and the X_ID reassigned bit (F_CTL bit 15) shall not be used.

25.3.3 Extended Link Services

If an N_Port performs any of the following Extended Link Service requests relating to an Exchange which has an Association Header, the Payload shall contain the current (first or final) Association Header.

Abort Exchange (ABTX)
Read Exchange Status (RES)
Reinstate Recovery Qualifier (RRQ)
Request Sequence Initiative (RSI)

The Association Header may be required to locate th Exchange Status Block since the state of X_ID validity may be unknown by the N_Port transmitting the request.

25.4 Error Recovery and other effects

When an N_Port uses a Process_Associator or Operation_Associator in order to better manage Exchange facilities, there are instances which restrict that management

a) unidirectional Exchanges do not allow the Responder (or Sequence Recipient) to invalidate and reassign its X_ID since it is unable to transmit an Association Header.
b) a Sequence Recipient is limited under the conditions in which it may invalidate its X_ID since the Sequence Initiator may not be able or willing to convey information such as a delay before the next Sequence will be initiated.
c) the Sequence Recipient is unable to invalidate its X_ID until it has received Sequence Initiative and transmitted the final Association Header.
d) in the Abort Sequence Protocol, a Recovery_Qualifier may be specified by the Recipient in the Basic Accept. The X_ID values in that Recovery_Qualifier shall be retired by both N_Ports for a timeout period of R_A_TOV in Class 2 and 3.

Additional factors in error recovery are introduced as a result of use of the Association Header. The additional factors arise from two different aspects of the Association Header procedure and protocol.

a) confirmation of Association Header transfer, and
b) confirmation of X_ID invalidation.

Both the confirmation of Association Header transfer and the confirmation of X_ID invalidation may be handled by performing the Abort Sequence Protocol. If the ABTS is rejected (BA_RJT) for an Invalid RX_ID-OX_ID combination, the N_Port transmitting the ABTS frame shall perform a Read Exchange Status request in order to determine the status of the X_IDs for the Exchange identified by the Association Header. If the ABTS is accepted, recovery proceeds normally.

25.5 Special F_CTL bits

The following F_CTL bits (15, 14) are explicitly used for X_ID reassignment and are only meaningful if X_ID reassignment and the Association Header are being used.

Bit 15 - X_ID reassigned

The X_ID reassigned bit shall be used at the beginning of a Sequence to indicate that either the Sequence Initiator, Recipient, or both are assigning new X_ID fields.

When the Sequence Initiator is reassigning its X_ID, it shall set this bit to one on the first Data frame of a Sequence and each successive Data frame transmitted until the first ACK for the Sequence has been received.

When the Sequence Recipient is reassigning its X_ID, X_ID interlock is required and it shall set this bit to one on the ACK to the first Data frame of the Sequence. A new X_ID shall only be assigned at the beginning of a Sequence. If the X_ID reassigned bit is set to zero by either the Initiator or Recipient, it indicates that its current X_ID assignment is valid.

NOTE - The X_ID reassigned bit is present so that an N_Port is able to determine that a change in X_ID value is taking place at a time other than the first Sequence of an Exchange, which is predictable. Otherwise, the N_Port could P_RJT the frame.

Bit 14 - Invalidate X_ID

Invalidate X_ID may be indicated by either the Sequence Initiator or Recipient if X_ID reassignment is identified as required in its Login parameters. An N_Port shall not invalidate its X_ID until it has confirmed successful transmission of its Operation_Associator in an Association Header.

Invalidate X_ID may only be indicated by the Sequence Initiator in the last Data frame of a Sequence (End_Sequence = 1). When the Sequence Initiator indicates that the current X_ID is being invalidated, or the Continue Sequence Condition bits are set to 1 1, the Sequence Recipient may also unassign its X_ID by setting the Invalidate X_ID bit to 1 in the ACK frame corresponding to the last Data frame of the Sequence.

If Sequence Initiative is passed, the Sequence Recipient may invalidate its X_ID by setting the Invalidate X_ID bit to 1 in the corresponding ACK frame. X_ID invalidation shall only occur at the end of a Sequence and shall not require transfer of Sequence Initiative at the same time.

X_ID invalidation by an N_Port shall require use of an Association Header as described in 19.4 and 25.3.1.

26 Flow control management

26.1 Introduction

Flow control is the FC-2 control process to pace the flow of frames between N_Ports and between an N_Port and the Fabric to prevent overrun at the receiver. Flow control is managed using end-to-end Credit, end-to-end Credit_CNT, ACK (ACK_1 or ACK_N), buffer-to-buffer Credit, buffer-to-buffer Credit_CNT, and R_RDY along with other frames.

Flow control is managed between N_Ports (end-to-end) and between N_Port and F_Port (buffer-to-buffer). Flow control management has variations dependent upon the Class of Service.

Applicability
Class 1 frames use end-to-end flow control. Class 1/**SOFc1** frame and Class 2 frames use both end-to-end and buffer-to-buffer flow controls. Class 3 uses only buffer-to-buffer flow control. Table 105 shows the applicability of the flow control mechanisms to each Class.

26.2 Physical flow control model

The physical flow control model is illustrated in figure 71. The model consists of following physical components:

- Each N_Port with Class 1 and Connectionless receive buffers.
- Each F_Port to which an N_Port is attached, with its receive buffers for Connectionless service. (Class 1 buffers internal to Fabric used for Class 1 service and Intermix are transparent to FC-2 flow control.)

Buffer participation
Buffering and transmission of Class 1 frames through the Fabric is transparent to FC-2. Class 1 buffering requirements during Intermix are specified in 22.4. The use of Class 1 buffers by the Fabric during Intermix is transparent to flow control. Connectionless buffers shall be shared by Class 2, Class 3 and Class 1/**SOFc1** frames. Table 106 summarizes the use of buffers for end-to-end and buffer-to-buffer flow controls.

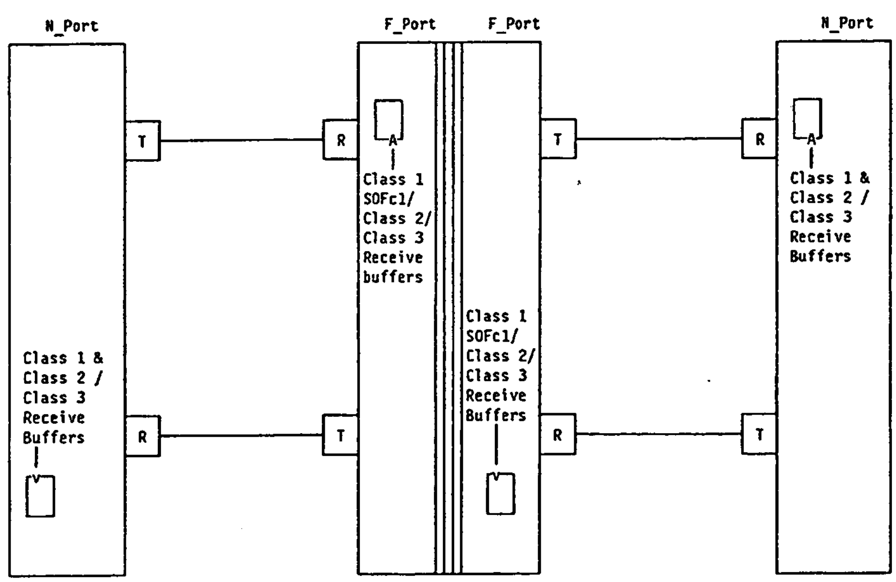


Figure 71 - Physical flow control model

Table 105 - Flow control applicability				
Flow Control methodology and mechanism	**Class 1 without SOFc1**	**Connect request frame with SOFc1**	**Class 2**	**Class 3**
End-to-end	Yes	Yes	Yes	No
Buffer-to-buffer	No	Yes	Yes	Yes
ACK_1 or ACK_N	Yes	Yes	Yes	No
ACK_0	One per Sequence	Yes	One per Sequence	No
R_RDY	No	Yes	Yes	Yes
F_BSY (DF)	No	Yes	Yes	No
F_BSY (LC)	No	No	Yes	No
F_RJT	No	Yes	Yes	No
P_BSY	No	Yes	Yes	No
P_RJT	Yes	Yes	Yes	No
Note: F_BSY (DF) indicates F_BSY response to a Data frame; F_BSY (LC) indicates F_BSY response to a Link_Control frame except P_BSY				

Table 106 - Buffer participation		
Participating buffers	**End to end flow control**	**Buffer to buffer flow control**
Class 1 buffers 1)	Class 1 frames	-
Connectionless buffers	Class 2 frames	Class 2, Class 3 and Class 1/SOFc1 frames
Note: 1) Participation of Class 1 buffers in the Fabric during Intermix is transparent to flow control.		

26.3 Credit and Credit_Count

Credit is the number of receive buffers allocated to a transmitting Port (an N_Port or an F_Port). Two types of Credits used in flow control are:

a) End-to-end Credit (EE_Credit)
b) Buffer-to-buffer Credit (BB_Credit)

The Credit_Count is managed by th Sequence Initiator.

Credit_Count manag ment
The Credit_Count management is internal to a Port and is transparent to the other Port. The Credit_Count may therefore be managed by the Port either by increasing or decreasing starting with an appropriate value.

Management by increasing the Credit_Count
This is the method specified in FC-PH. In this method, the Credit_Count is initialized to a 0 (zero) value and increased by a specified value (see tables 107, 108, and 109) to a maximum value given by the Credit. The Credit_Count in this case represents the number of outstanding Data frames which have not been acknowledged by the Sequence Recipient. The Credit_Count is a positive integer and shall not exceed the value of Credit to avoid the possibility of overflow at the receiver. The Credit_Count shall not be decreased below 0 (zero).

Management by decreasing the Credit_Count
This method is not specified in this document but may be used in a given implementation. In this method, the Credit_Count is initialized to the value of the Credit and decreased to the minimum value of 0 (zero). The Credit_Count in this case represents the number of Data frames that may be sent without the possibility of causing overflow at the rec iver. The Credit_Count is a positive integer and shall not be decreased below 0 (zero).

Cr dit_Count types
Corresponding to two types of Credits listed above, two types of Credit_Counts used are:

a) End-to-end Credit_Count (EE_Credit_CNT)
b) Buffer-to-buffer Credit_Count (BB_Credit _CNT)

Usage
The N_Port transmitting Class 1 or Class 2 Data frames shall use the end-to-end Credit allocated by the receiving N_Port for end-to-end flow control and manage the corresponding end-to-end Credit_Count. Class 3 Data frames do not participate in end-to-end flow control. When a Port (an N_Port or an F_Port) is transmitting Data frames or Link_Control frames to the attached Port, the transmitting Port shall use BB_Credit allocated by the receiving Port for buffer-to-buffer flow control and manage the corresponding BB_Credit_Count.

26.4 End-to-end flow control

End-to-end flow control is an FC-2 control process to pace the flow of frames between N_Ports. End-to-end flow control is used by an N_Port pair in Class 1 or Class 2.

End-to-end flow control is performed with EE_Credit_CNT with EE_Credit as the controlling parameter.

26.4.1 End-to-end management rules summary

End-to-end management rules are summarized for error free functioning. Management of EE_Credit_CNT is summarized in table 107. The EE_Credit recovery is specified in 26.4.9.

26.4.2 Sequence Initiator

a) The Sequence Initiator is responsible for managing EE_Credit_CNT across all active Sequences.
b) The Sequence Initiator shall not transmit a Data frame unless the allocated EE_Credit is greater than zero and the EE_Credit_CNT is less than this EE_Credit.
c) In Class 1 or Class 2, the valu of the EE_Credit_CNT is set to zero (0) at the end of N_Port Login, N_Port re-Login, or Link Credit Reset (s e 20.3.4).
d) In Class 1, the EE_Credit_CNT is set to one (1), on transmitting Class 1/**SOFc1** frame. It is incremented by one (1) for each subsequent Class 1 Data frame transmitted. In the case of ACK_0 usage, EE_Credit_CNT management is not applicable.
e) The EE_Credit_CNT is incremented by one (1) for each Class 2 Data frame transmitted. In the case of ACK_0 usage, EE_Credit_CNT management is not applicable.
f) The Sequence Initiator decrements the EE_Credit_CNT by a value of one for each ACK_1 (parameter field: History bit = 1, ACK_CNT = 1), F_BSY(DF), F_RJT, P_BSY, or P_RJT received.
g) For an ACK_1 (parameter field: History bit = 0, ACK_CNT = 1) received, the Sequence Initiator decrements the EE_Credit_CNT by a value of one for the current SEQ_CNT in the ACK_1 plus all unacknowledged Data frames with lower SEQ_CNTs. If any of these ACKs with lower SEQ_CNT is received later, it is ignored and Credit_Count is not decremented.
h) For an ACK_N (parameter field: History bit = 1, ACK_CNT = N) received, the Sequence Initiator decrements the EE_Credit_CNT by a value of N.
i) For an ACK_N (parameter field: History bit = 0, ACK_CNT = N) received, the Sequence Initiator decrements the EE_Credit_CNT by a value of N plus all unacknowledged Data frames with lower SEQ_CNTs. If any of these ACKs with lower SEQ_CNT is received later, it is ignored and Credit_Count is not decremented.
j) For an ACK_0 (parameter field: History bit = 0, ACK_CNT = 0) received, the Sequence Initiator recognizes that the Sequence has been received successfully or unsuccessfully, or that the interlock is being completed (see 20.3.2 and 20.3.2.2), but does not perform any EE_Credit_CNT management.
k) For an ACK_1 or ACK_N with EOFt or EOFdt received, even if the History bit contains a value of 1, the Sequence Initiator shall recover the Credit for the Sequence by decrementing the EE_Credit_CNT by an additional value equal to all unacknowledged Data frames with lower SEQ_CNT of the Sequence. If any of these ACKs with lower SEQ_CNT is received later, it is ignored and Credit_Count is not decremented.

26.4.3 Sequence R cipient

a) The Sequence Recipient is responsible for acknowledging valid Data frames received (see 20.3.2.2).

b) The Sequence Recipient is allowed to use ACK_0, ACK_1, or ACK_N as determined during N_Port Login (see 23.6). The Sequence Recipient rules for using ACK_0, ACK_1, or ACK_N are different and are listed for a non-streamed Sequence first, followed by additional rules for streamed Sequences.

26.4.3.1 ACK_0

If ACK_0 is used (see 20.3.2 and 20.3.2.2), the following rules apply to the Sequence Recipient:

a) ACK_0 shall not participate in end-to-end flow control.

b) A single ACK_0 per Sequence shall be used to indicate successful or unsuccessful Sequence delivery at the end of the Sequence except under specified conditions (see 20.3.2 and 20.3.2.2).

c) Both the History bit and the ACK_CNT of the Parameter field shall be set to 0 (see 20.3.2.2).

d) The ACK_0 used at the end of a Sequence shall have the End_Sequence bit set to 1. The ACK_0 used at the end of a Sequence shall be ended with EOFt or EOFdt in Class 1 and with EOFt in Class 2.

26.4.3.2 ACK_1

If ACK_1 is used, the following rules apply to the Sequence Recipient:

a) For each valid Data frame acknowledged an ACK_1 shall be sent with ACK_CNT set to 1.

b) The History bit of the Parameter field shall be set to 1 if at least one ACK is pending for a previous SEQ_CNT for the Sequence, or shall be set to 0 if no ACK is pending for any previous SEQ_CNT for the Sequence (see 20.3.2.2).

c) In Class 1, the Sequence Recipient shall withhold transmission of the last ACK_1 until all preceding ACK_1s corresponding to all Data frames with previous SEQ_CNTs have been transmitted. The last ACK_1 in Class 1 shall have the End_Sequence bit set to 1, History bit set to 0 and shall contain EOFt or EOFdt.

d) In Class 2, the last ACK_1 shall be issued by the Sequence Recipient in one of the two ways specified.

1) In Class 2 the Sequence Recipient shall withhold transmission of the last ACK_1 until all preceding Data frames with lower SEQ_CNTs have been received, processed, and corresponding ACK_1s transmitted (see 24.3.10). In this case, the last ACK_1 transmitted by the Sequence Recipient shall have the End_Sequence bit set to 1, History bit set to 0 and shall contain EOFt.

2) In Class 2, in response to the last Data frame (End_Sequence bit = 1) transmitted by the Sequence Initiator, if any of the Data frame is pending for the Sequence, the Sequence Recipient is allowed to transmit ACK_1 (with End_Sequence bit set to 0) but with EOFn in lieu of EOFt. In this case, the last ACK_1 transmitted by the Sequence Recipient shall have the End_Sequence bit set to 1, History bit set to 1 and shall contain EOFt.

26.4.3.3 ACK_N

If ACK_N is used, the following rules apply to the Sequence Recipient:

a) Each ACK_N shall specify the number of consecutive Data frames acknowledged collectively (N) in ACK_CNT of the Parameter field. The History bit of Parameter field shall be set to 1, if at least one ACK is pending for a frame with a lower SEQ_CNT than any of the frames being acknowledged or shall be set to 0 if no ACK is pending for a frame with a lower SEQ_CNT than any of the frames being acknowledged (see 20.3.2.2).

b) The value of ACK_CNT for each ACK_N is implementation dependent.

c) Each ACK_N shall specify in SEQ_CNT field, the highest SEQ_CNT of the consecutive Data frames being acknowledged.

d) An ACK_N shall be sent on detection of a missing Data frame to acknowledge Data frames received with SEQ_CNTs lower than the SEQ_CNT of the missing frame.

e) In Class 1, the Sequence Recipient shall withhold transmission of the last ACK_N until all preceding ACK_Ns corresponding to all Data frames with lower SEQ_CNTs hav been transmitted. The last ACK_N in Class 1 shall

have the End_Sequence bit set to 1, History bit set to 0 and shall contain EOFt or EOFdt.

f) In Class 2, the last ACK_N shall be issued by the Sequence Recipient in one of the two ways specified.

1) In Class 2 the Sequence Recipient shall withhold transmission of the last ACK_N until all Data frames with SEQ_CNTs lower than those which are acknowledged in the last ACK_N, have been received, processed, and corresponding ACK_Ns transmitted (see 24.3.10). In this case, the last ACK_N transmitted by the Sequence Recipient shall have the End_Sequence bit set to 1, History bit set to 0 and shall contain EOFt.

2) In Class 2, in response to the last Data frame (End_Sequence bit = 1) transmitted by the Sequence Initiator, if any of the Data frame is pending for the Sequence, the Sequence Recipient is allowed to transmit ACK_N (with End_Sequence bit set to 0) but with EOFn in lieu of EOFt. In this case, the last ACK_N transmitted by the Sequence Recipient shall have the End_Sequence bit set to 1, History bit set to 1 and shall contain EOFt.

g) ACK_N is used for functions other than flow control. These functions are listed in 20.3.2.2.

h) The maximum value for ACK_CNT (the largest number of consecutive frames acknowledged collectively) chosen by the Sequence Recipient shall be less than the EE_Credit.

NOTE - If the maximum value for N and the EE_Credit are equal and a frame is lost, the receiver will be waiting for the frame and the sender will be waiting for the ACK resulting in the Sequence timeout condition. Keeping N less than Credit helps prevent the Sequence timeout condition unless multiple frames (greater than Credit minus N) are lost. Implementers are cautioned that keeping N less than Credit does not help the elimination of the Sequence timeout condition if Credit is collectively managed for all active Sequences. The Sequence timeout condition is detected through timeout and rectified through Credit recovery (see 29.2.4 and 26.4.9).

(See ACK_N usage examples in annex O.)

26.4.3.4 Last ACK time ut

If a Sequence error is detected or the E_D_TOV expires when the Sequence Recipient is withholding the last ACK for a Sequence and waiting to send other ACKs for that Sequence, the Sequence Recipient supporting discard policy shall set Abort Sequence bits and transmit the last ACK (see 24.6.5). The Sequence Recipient supporting the Process Policy shall transmit the last ACK without setting the Abort Sequence bits (see 24.3.10.4).

26.4.3.5 Streamed Sequences

a) Each of the streamed Sequences shall follow all the rules for a non-streamed Sequence.

b) In addition, in the case of multiple Sequence discard policy, the last ACK for the succeeding Sequence shall be withheld until all the previous Sequences are complete and deliverable. This additional withholding, for previous Sequences to complete and be deliverable, is not applicable to the case of Single Sequence discard policy.

26.4.4 EE_Credit

EE_Credit is the number of receive buffers in the Sequence Recipient that have been allocated to a given Sequence Initiator. EE_Credit represents the maximum number of unacknowledged or outstanding frames that can be transmitted without the possibility of overrunning the receiver at the Sequence Recipient. EE_Credit is defined per Class per Sequence Recipient and managed by the Sequence Initiator. Class 1 EE_Credit represents the number of Class 1 receive buffers and Class 2 EE_Credit represents the number of Class 2 buffers allocated to the Sequence Initiator. EE_Credit is not applicable to Class 3. The value of EE_Credit allocated to the Sequence Initiator is conveyed to this N_Port through the EE_Credit field of the Service Parameters. The minimum or default value of EE_Credit is one (1).

EE_Credit is used as a controlling parameter in end-to-end flow control.

26.4.5 EE_Credit_C unt

EE_Credit_CNT is defined as the number of unacknowledged or outstanding frames awaiting a response and represents the number of receive buffers that are occupied at the Sequence Recipient. To track the number of frames transmitted and outstanding, the Sequence Initiator uses the above variable.

26.4.6 EE_Credit management

EE_Credit management involves an N_Port establishing and revising EE_Credit with the other N_Port it intends to communicate with, for Class 1 or Class 2 or both.

N_Port Login is used to establish and optionally revise these EE_Credit values. The Estimate Credit procedure may be used to estimate and revise End-to-End Credit for streaming. The Advise Credit Sequence and associated Accept Sequence may also be used as a stand alone procedure to revise the EE_Credit (see 22.4). The Service Parameters interchanged during N_Port Login provide the Class 1 or Class 2 EE_Credit separately in their respective Credit fields.

EE_Credit is obtained by a Sequence Initiator during N_Port Login from the N_Port to which it is logging in. EE_Credit allocated by the Sequence Recipient forms the maximum limit for the EE_Credit_CNT value. The EE_Credit_CNT value is set at zero (0), at the end of initialization, Login or re-Login. The EE_Credit_CNT is incremented, decremented or left unaltered as specified by the flow control management rules (see 26.4.1). The EE_Credit_CNT shall not exceed the EE_Credit value to avoid possible overflow at the receiver.

The Sequence Initiator shall allocate the total N_Port Credit associated with a Sequence Recipient among all active Sequences associated with that Sequence Recipient. The Sequence Initiator function may dynamically alter the Credit associated with each active Sequence as long as the total N_Port Credit specified for the Sequence Recipient is not exceeded. In the event of an abnormal termination of a Sequence using the Abort Sequence Protocol, the Sequence Initiator may reclaim the Sequence Credit allocation when the Accept response has been received to the Abort Sequence frame.

The N_Port is responsible for managing EE_Credit_CNT using EE_Credit as the upper bound on a per N_Port basis.

26.4.7 End-to-end flow control model

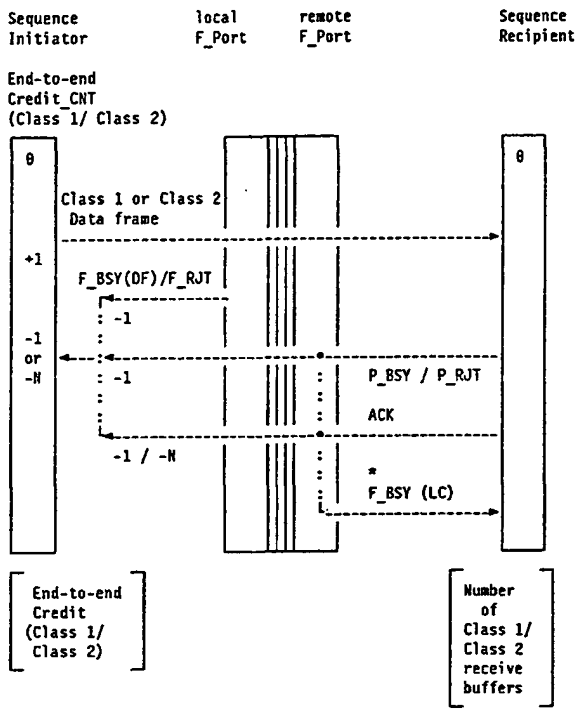
The end-to-end flow control model is illustrated in figure 72. The model includes flow control parameters, control variables and resources for a Data frame from a Sequence Initiator and ACK_1 or ACK_N or BSY/RJT in response from the Sequence Recipient.

— The Sequence Recipient provides a number of Class 1 and/or Class 2 receive buffers.

— The Sequence Initiator obtains the allocation of Class 1 and/or Class 2 receive buffers, as Class 1 and Class 2 EE_Credits, respectively. That allocation is distributed among all the active Sequences for a specific Sequence Recipient.

— The Sequence Initiator manages the end-to-end flow by managing Class 1 and Class 2 end-to-end Credit-CNT(s). (That management is distributed among all the active Sequences for a specific Sequence Recipient.)

The model illustrates all possible replies to the Data frame. The EE_Credit_CNT is decremented by one or N depending upon the type of Link_Control frame received.



Legend:
N = number of valid Data frames acknowledged collectively
* F_BSY(LC) is a valid response only to Class 2

Figure 72 - End-to-end flow control model

26.4.8 End-to-end Class dependency

Allocation of EE_Credit and management of EE_Credit_CNT have some variations dependent upon Class of Service.

End-to-end Credit allocation

– Each Sequence Recipient may allocate the same Class 1 N_Port Credit value to each N_Port it is logging into. This Class 1 Credit value may be the maximum supportable by the Sequence Recipient.

– Each Sequence Recipient allocates some number of its receive buffers for Class 2 Service to N_Ports it is logging into. The sum of allocated Class 2 buffers may exceed the total number of Class 2 buffers supported at the Sequence Recipient. This excess buffer allocation shall not result in overrun. Class 2 EE_Credit allocation depends upon system r quirements which are outside the scope of FC-PH.

EE_Credit_CNT management

– Since Class 2 supports demultiplexing to multiple Sequence Recipient, the Sequence Initiator manages a Connectionless EE_Credit_CNT for each Sequence Recipient currently active, with that Sequence Recipient's EE_Credit as the upper bound.

– A Class 3 N_Port does not perform EE_Credit_CNT management.

26.4.9 EE_Credit recovery

a) In Class 1 and Class 2, EE_Credit can be recovered by Sequence Initiator when a Sequence is terminated, by the number of unacknowledged Data frames associated with the Sequence being terminated. Termination may be normal or abnormal.

b) In Class 1 and Class 2, EE_Credit may be recovered by the Sequence Initiator within the Sequence by detection of SEQ_CNT discontinuity in ACK, if the ACK received contains zero in the History bit of the Parameter field. Otherwise, EE_Credit can be recovered by the Sequence Initiator at the termination of the Sequence.

If an ACK is received which contains a zero in the History bit of the parameter field, then EE_Credit for any outstanding ACK with lower SEQ_CNT shall be reclaimed. If any of these outstanding ACKs arrive later due to misordering, they shall not affect the EE_Credit_Count.

c) Class 1 EE_Credit is also recovered when a Dedicated Connection is removed by either **EOFdt** or by the Link Reset Protocol.

d) In Class 1 and Class 2, EE_Credit may also be recovered by an N_Port through re-Login.

e) Class 2 EE_Credit may also be reinitialized to N_Port Login value using Link Credit Reset (see 20.3.4).

26.5 Buffer-to-buffer flow control

Buffer-to-buffer flow control is an FC-2 staged control process to pace the flow of frames. The buffer-to-buffer control occurs in both directions between

– Sequence Initiator and local F_Port
– remote F_Port and Sequence Recipient N_Port
– Sequence Initiator and Sequence R cipient N_Ports in point-to-point topology

26.5.1 Buffer-to-buff r manag ment rules summary

Buffer-to-buffer flow control rules are summarized. Managing BB_Credit_CNT at an N_Port or an F_Port is summarized in table 108.

a) Each Port (N_Port or F_Port) is responsible for managing the BB_Credit_CNT.
b) The sending N_Port or F_Port shall not transmit a Class 2, Class 3, or Class 1/SOFc1 frame unless the allocated BB_Credit is greater than zero and the BB_Credit_CNT is less than this BB_Credit. To avoid possible overrun at the receiver, each port manages BB_Credit_CNT less than BB_Credit.
c) Each Port sets BB_Credit_CNT value to zero (0) at the end of Fabric Login, or Fabric re-Login.
d) Each Port increments BB_Credit_CNT by one (1) for each Class 2, Class 3, or Class 1/SOFc1 frame transmitted and decrements by one (1) for each R_RDY received.
e) Each Port issues an R_RDY for each Class 2, Class 3, or Class 1 frame with SOFc1 received (see 20.3.2.1).

26.5.2 BB_Credit

BB_Credit represents the number of receive buffers supported by a Port (N_Port or F_Port) for receiving Class 1/SOFc1, Class 2, or Class 3 frames. BB_Credit values of the attached ports are mutually conveyed to each other during the Fabric Login through the Credit field of common Service Parameters (see 23.6). The minimum or default value of BB_Credit is one (1).

BB_Credit is used as the controlling parameter in buffer-to-buffer flow control.

26.5.3 BB_Credit_Count

BB_Credit_CNT is defined as the number of unacknowledged or outstanding frames awaiting R_RDY responses from the directly attached port. It represents the number of receive buffers that are occupied at the attached port. To track the number of frames transmitted for which R_RDY respons s are outstanding, the transmitting Port uses the above variable.

26.5.4 BB_Credit management

BB_Credit management involves a Port receiving the BB_Credit value from the Port to which it is directly attached. Fabric Login is used to accomplish this. The common Service Parameters interchanged during Fabric Login provide these values (see 23.6 and 23.7).

The transmitting Port is responsible to manage BB_Credit_CNT with BB_Credit as its upper bound.

26.5.5 Buffer-to-buffer flow control model

The Buffer-to-buffer flow control model is illustrated in figure 73. The model includes flow control parameters, control variables for a Class 2, Class 3, or Class 1/SOFc1 frame and R_RDY as its response, and the resources for Connectionless service. All possible responses to a Class 2 or Class 3 Data frame are illustrated.

— Each N_Port and F_Port provides a number of receive buffers for Connectionless service.

— Each N_Port obtains the allocation of receive buffers from the F_Port (or N_Port in case of point-to-point topology) to which it is attached, as BB_Credit. Each F_Port obtains the allocation of receive buffers from the N_Port to which it is attached, as total BB_Credit for Connectionless service.

— Each Port manages the buffer-to-buffer flow by managing the BB_Credit_CNT for the Connectionless service, with BB_Credit as the upper limit.

26.5.6 Buffer-to-buffer Class dependency

Allocation of BB_Credit and management of BB_Credit_CNT for Connectionless service are described.

BB_Credit allocation
Each Port allocates the total number of Connectionless buffers to the Port to which it is directly attached.

BB_Credit_CNT manag ment
A Port manages the BB_Credit_CNT with BB_Credit as the upper bound.

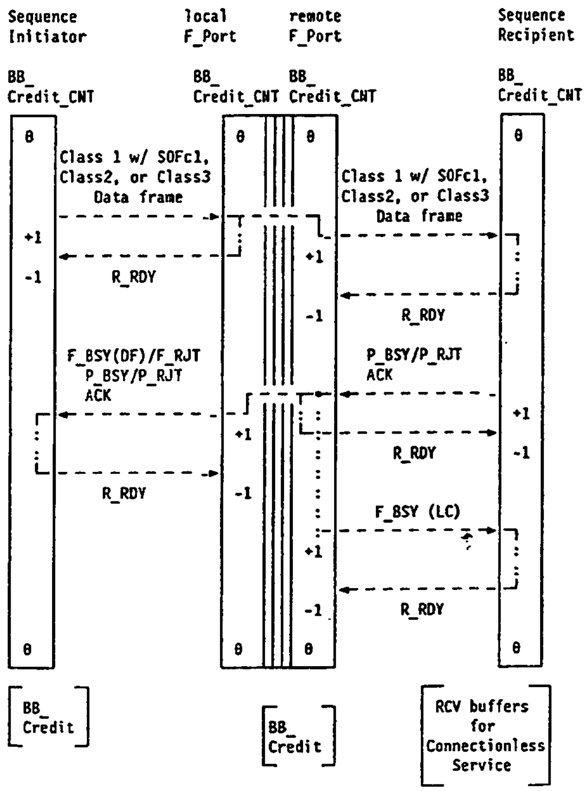


Figure 73 - Buffer-to-buffer flow control model

26.5.7 Class dependent frame flow

The Class dependent flow of frames accomplishing buffer-to-buffer flow control for the following cases are illustrated:

- Class 1/SOFc1 frame with delivery or non-delivery to the Fabric (see figure 74).
  - Possible responses from the F_Port or an N_Port within the Fabric to a Class 1/SOFc1 frame are illustrated.
- Class 1/SOFc1 frame with delivery or non-delivery to an N_Port (see figure 75).
  - Possible responses from the F_Port and the destination N_Port to a Class 1/SOFc1 frame are illustrated.
- Class 2 with delivery or non-delivery to the Fabric (see figure 76).
  - Possible responses from the F_Port or an N_Port within the Fabric to a Class 2 are illustrated.
- Class 2 with delivery or non-delivery to an N_Port (see figure 77).
  - Possible responses from the F_Port and the destination N_Port to a Class 2 Data frame are illustrated.
- Class 3 (see figure 78).
  - Possible responses from the F_Port and the destination N_Port to a Class 3 Data frame are illustrated.

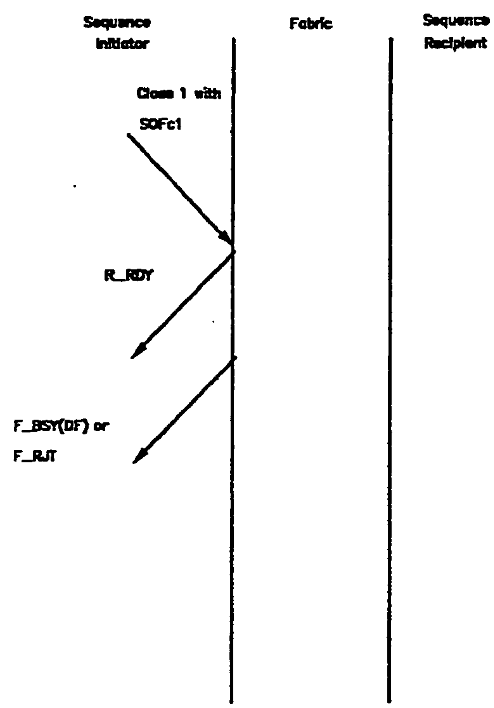


Figure 74 - Class1/SOFc1 frame flow with delivery or non-delivery to the Fabric

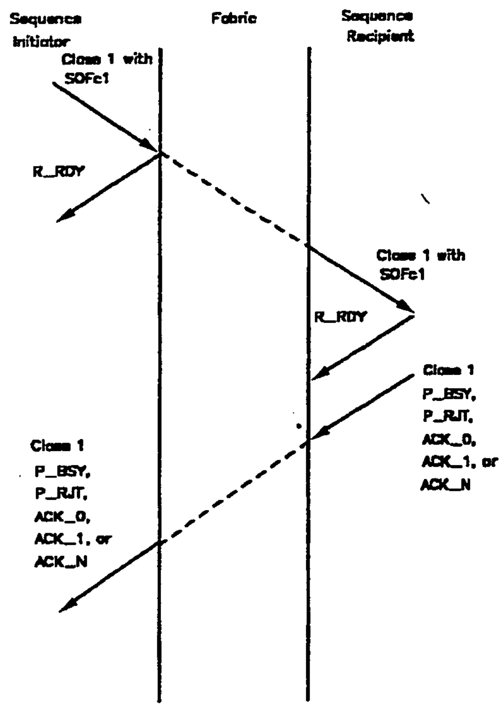


Figure 75 - Class 1/SOFc1 frame flow with delivery or non-delivery to an N_Port

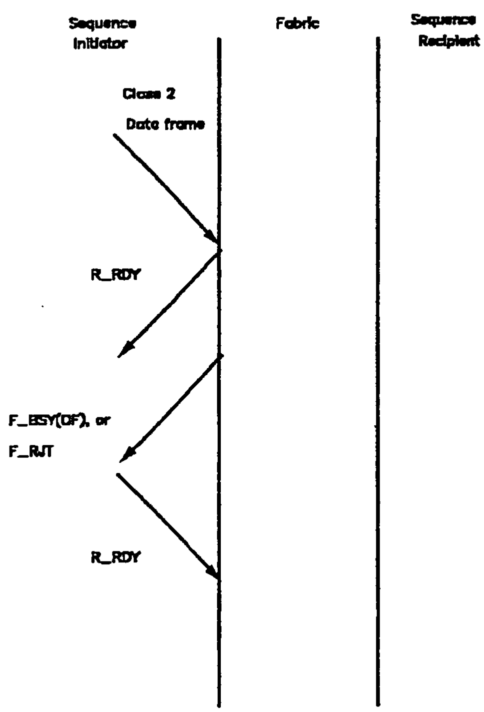


Figure 76 - Buffer-to-buffer - Class 2 frame fl w with delivery r non-d livery to the Fabric

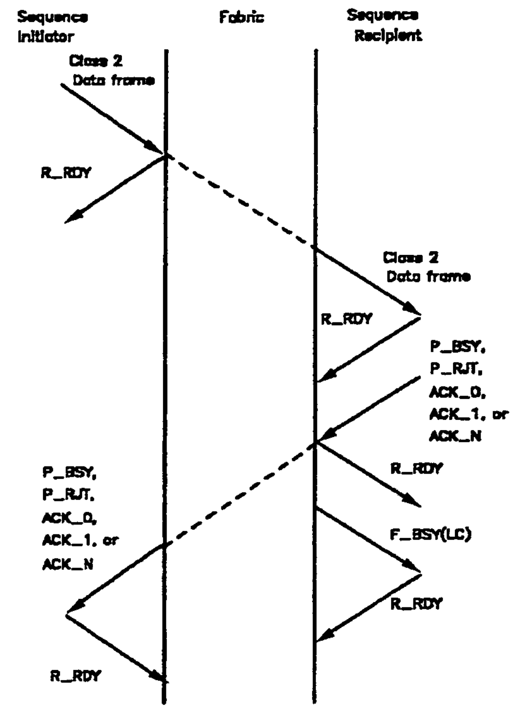


Figure 77 - Buffer-to-buffer - Class 2 frame flow with delivery or non-delivery to an N_Port

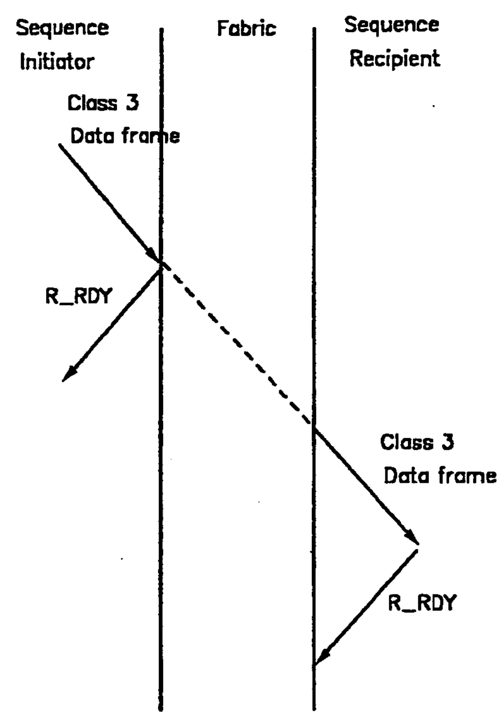


Figure 78 - Buffer-to-buff r - Class 3 frame flow

26.5.8 R_RDY

R_RDY is the pacing mechanism exclusively used for buffer-to-buffer flow control. For any Class 2, Class 3, or Class 1/**SOFc1** frame received at an F_Port or at an N_Port, each Port issues an R_RDY primitive.

26.5.9 BB_Credit_Count reset

BB_Credit_Count is reinitialized to login value for both N_Port and F_Port on Fabric re-Login or after the Link Reset Protocol has been performed.

26.6 BSY / RJT in flow control

In Class 1 end-to-end flow control, neither F_BSY, F_RJT, nor P_BSY occurs, except for a Class 1/**SOFc1**. In Class 2 end-to-end flow control, F_BSY, F_RJT, P_BSY or P_RJT may occur for any Data frame. Each of these responses contributes to end-to-end and buffer-to-buffer flow controls.

End-to-end Class 2 buffers at the Sequence Recipient N_Port are shared by multiple source N_Ports which may be multiplexing Data frames. This Class 2 multiplexing requires the distribution of Class 2 Credit to each source N_Port to be honored to prevent BSY or RJT. Unless an adequate number of end-to-end Class 2 buffers is available and Credit distribution is honored, a BSY or RJT may occur in Class 2. If buffer-to-buffer flow control rules are not obeyed by the transmitter, the results are unpredictable e.g., if a Class 2 frame is received without Credit and the receiver does not have a buffer to receive it, the receiver may discard the frame without issuing a P_BSY or P_RJT.

26.7 LCR in flow c ntrol

LCR does not need EE_Credit and does not participate in end-to-end flow control. LCR participates only in buffer-to-buffer flow control as Class 2. (see figure 79). In response to an LCR, the Fabric shall issue an R_RDY and may issue a F_BSY or F_RJT. The destination N_Port shall issue an R_RDY and may issue a P_RJT (see 20.3.4). The destination N_Port shall not issue a P_BSY to an LCR.

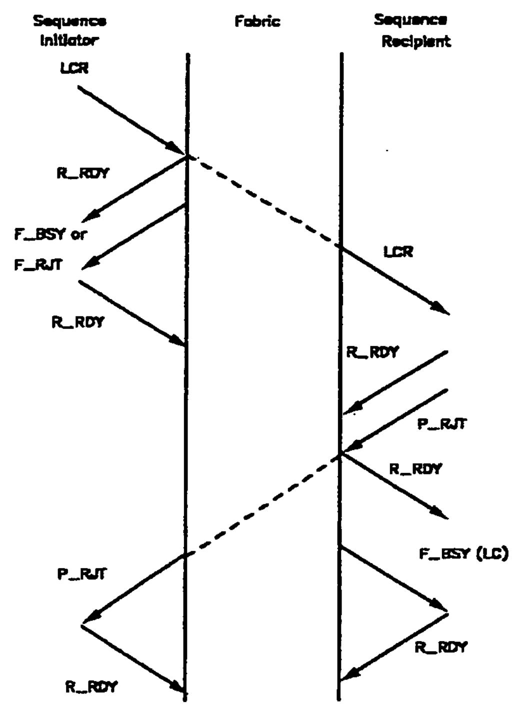


Figure 79 - LCR frame flow and possible responses

Flow control model for an LCR frame is illustrated in figure 80 which covers the buffer-to-buffer flow control for all possible responses to an LCR.

EE_Credit recovery
After issuing the LCR, the Sequence Initiator shall reinitialize its EE_Credit to N_Port Login value for the destination N_Port. The Sequence Initiator shall not wait for any response before reinitializing this Credit (see 20.3.4).

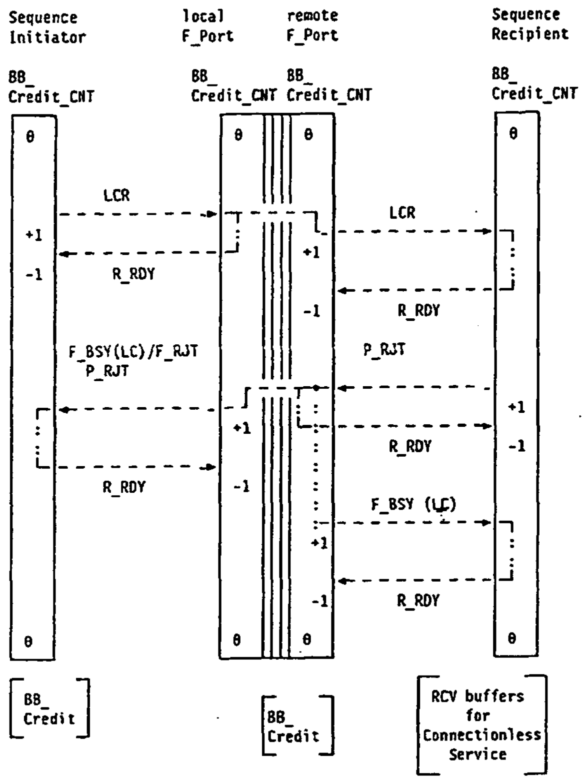
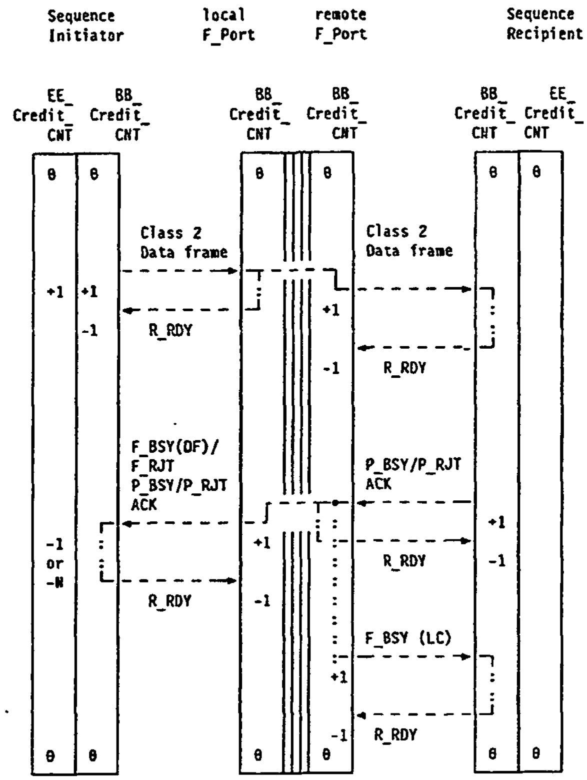


Figure 80 - LCR flow control model

26.8 Integrated Class 2 flow control

Integrated buffer-to-buffer and end-to-end flow controls applicable to Class 2 is illustrated in figure 81 for a Data frame from the Sequence Initiator and its response from the Sequence Recipient.

Management
Integrated Class 2 flow control management is summarized in table 109.



N = number of valid frames acknowledged collectively

Figure 81 - Integrated Class 2 flow control

26.9 Intermix

The flow control model for Intermix is represented by the superposition of end-to-end and Buffer-to-buffer flow controls (figures 72 and 73).

26.10 Point-to-point topology

All the flow control models specified above in this clause apply to Fabric topology. The flow control model for Point-to-point topology is represented by the corresponding model for the Fabric topology, without the flow of F_BSY(DF), F_BSY(LC), and F_RJT.

Table 107 - End-to-end flow control management	
Activity	**EE_Credit_CNT (N_Port only)**
N_Port transmits a Class 1 or Class 2 Data frame	+1
N_Port transmits an LCR	And reinitializes End-to-end Credit for the destination N_Port to its Login value.
N_Port receives F_BSY(DF), F_RJT, P_BSY, or P_RJT	-1
N_Port receives F_BSY(LC)	NA
N_Port receives ACK_1 (Parameter field: History bit = 1, ACK_CNT = 1)	-1
N_Port receives ACK_1 (Parameter field: History bit = 0, ACK_CNT = 1)	subtract 1 for current SEQ_CNT of the ACK_1 and also subtract all unacknowledged lower SEQ_CNTs (see 20.3.2.2)
N_Port receives ACK_N (Parameter field: History bit = 1, ACK_CNT = N)	-N
N_Port receives ACK_N (Parameter field: History bit = 0, ACK_CNT = N)	subtract N as indicated in ACK_CNT and also subtract all unacknowledged lower SEQ_CNTs (see 20.3.2.2)
N_Port receives ACK_0 (Parameter field: History bit = 0, ACK_CNT = 0)	NA (see 20.3.2.2)
N_Port receives a Data frame (Class 1, Class 2, or Class 3)	NA
N_Port receives an LCR	NA*
N_Port transmits a Class 3 Data frame	NA
N_Port transmits P_BSY or P_RJT	NA
N_Port transmits ACK	NA

Note: NA - Not Applicable
N - number of valid frames acknowledged collectively
* On receipt of LCR, the Sequence Recipient resets buffer (see 20.3.4)

Table 108 - Buffer-to-buffer flow control management	
Activity	**BB_Credit_CNT (N_Port or F_Port)**
N_Port or F_Port transmits any Class 2, Class 3, Class 1 frame with SOFc1 (including F_BSY(DF), F_BSY(LC), F_RJT, P_BSY, P_RJT or LCR)	+1
N_Port or F_Port receives R_RDY	-1
N_Port or F_Port receives any Class 2, Class 3, or Class 1 frame with SOFc1 (including F_BSY(DF), F_BSY(LC), F_RJT, P_BSY, P_RJT or LCR)	NA
N_Port or F_Port transmits R_RDY	NA

Note: NA - Not Applicable

Table 109 - Integrated Class 2 flow control management

Activity	N_Port		F_Port
	EE_Credit_CNT	BB_ Credit_CNT	BB_ Credit_CNT
Port transmits a Class 2 Data frame	+1	+1	+1
Port transmits an LCR	Reinitializes to Login value	+1	+1
Port receives R_RDY	NA	-1	-1
Port receives F_BSY(DF), F_RJT, P_BSY, or P_RJT	-1	NA	NA
Port receives F_BSY(LC)	NA	NA	NA
Port receives ACK_1 (Parameter field: History bit=1, ACK_CNT=1)	-1	NA	NA
Port receives ACK_1 (Parameter field: History bit=0, ACK_CNT=1)	subtract 1 for current SEQ_CNT of the ACK_1 and also subtract all unacknowledged lower SEQ_CNTs (see 20.3.2.2)	NA	NA
Port receives ACK_N (Parameter field: History bit=1, ACK_CNT=N)	-N	NA	NA
Port receives ACK_N (Parameter field: History bit=0, ACK_CNT=N)	subtract N as indicated in ACK_CNT and also subtract all unacknowledged lower SEQ_CNTs (see 20.3.2.2)	NA	NA
Port receives ACK_0 (Parameter field: History bit=0, ACK_CNT=0)	NA (see 20.3.2.2)	NA	NA
Port receives an LCR	NA *	NA	NA
Port receives a Class 2 Data frame	NA	NA	NA
Port transmits R_RDY	NA	NA	NA
Port transmits F_BSY, F_RJT, P_BSY, P_RJT, or ACK	NA	+1	+1

Note: NA - Not Applicable
N - number of valid frames acknowledged collectively
* On receipt of LCR, the Sequence Recipient resets buffer (see 20.3.4)

27 Segmentation and reassembly

27.1 Introduction

Segmentation and reassembly are the FC-2 functions provided to subdivide application data corresponding to an Information Category to be transferred into Payloads, compute Relative Offset values, embed each Payload and the corresponding Relative Offset value in an individual frame, transfer these frames over the link, and reassemble at the receiving end the application data of the Information Category as sent by the sending end.

Application data mapping
Mapping application data to Upper Level Protocols (ULPs) is outside the scope of FC-PH. ULPs maintain the status of application data transferred.

27.2 Sending end

An upper level (a level above FC-2) at the sending end shall define a Relative Offset space for each Information Category for which Relative Offset usage is supported. An upper level specifies a block or a collection of subblocks to be transferred within a Sequence.

27.2.1 Relative Offset space

The Relative Offset space shall start from zero, representing an upper level-defined origin, and extend to its highest value. Application data for an Information Category transferred may be mapped to all or portions of Relative Offset space.

27.2.2 Block

A block is an upper level construct of application data related to a single Information Category and transferred within a single Sequence. If Continuously Increasing Relative Offset is used, a block shall be specified. A block is allowed to map to all or part of Relative Offset space of the Information Category. An upper level shall specify an Initial Relative Offset (IRO_i) for each block. The Initial Relative Offs t value is allowed to be zero or non-zero.

27.2.3 Subblock

A subblock is an upper level construct which contains partial data for a single Information Category. A collection of subblocks is specified for a given Information Category to be transferred within a Sequence. The subblocks shall be specified, only if the Sequence Recipient supports Random Relative Offset. An upper level shall specify an Initial Relative Offset for each subblock. The Initial Relative Offset value is allowed to be zero or non-zero. The Initial Relative Offset values of subblocks of a given Information Category transmitted consecutively are allowed to be random.

The collection of subblocks for a single Information Category is allowed to map to portions or all of Relative Offset space for the Information Category.

27.2.4 Sequence

The blocks to be transferred within a single Sequence and the Information Category for each block shall be specified to FC-2 by an upper level. The collections of subblocks to be transferred within a Sequence and the information Category for each subblock shall be specified to FC-2 by an upper level.

27.2.5 Relationship between Sequences

The Relative Offset relationship between multiple Sequences of a given Information Category shall be specified by an upper level. This relationship is transparent to FC-2.

27.3 FC-2

Mechanisms
FC-2 mechanisms to support this function are

a) Relative Offset or SEQ_CNT
b) Sequence
c) F_CTL bit indicating the presence of Relative Offset in the parameter field (see 18.2)

Relative Offset
The Parameter field in the Frame_Header may be used to hold the value of Relative Offset which is the relative displacement of first byte of

Payload related to an upper level-defined origin for a given Information Category.

Sequence Count (SEQ_CNT)
If Relative Offset is not used for reassembly, the Information Category shall be reassembled using the SEQ_CNT. The reassembly shall be performed by joining or concatenating the Payloads of the Data frames in the continuously increasing order of SEQ_CNTs starting from the SEQ_CNT of the first Data frame to that of the last Data frame of a given Sequence.

For an Information Category i within a given Sequence, the reassembly is expressed as
Application $data_i$ =
PL_i (m) || PL_i (m+1) || ooo || PL_i (m+n)
where PL (m) represents Payload
of a frame with SEQ_CNT m
n is the total frame count
within the Sequence
m is the lowest SEQ_CNT (zero or non-zero) and m+n is the highest SEQ_CNT.

If the SEQ_CNT wraps to zero from hex FFFF within a Sequence, the reassembly shall be continued according to modulo 65536 arithmetic (i.e., SEQ_CNT = 0 follows SEQ_CNT = hex FFFF).

27.4 Login

The following Common Service Parameters related to segmentation and reassembly are interchanged during N_Port Login (see figure 61):

a) Relative Offset by Information Category
b) Continuously Increasing or Random Relative Offset

Through the interchange of these Login parameters, the Sequence Recipient indicates its Relative Offset requirements to the Sequence Initiator. The Sequence Recipient indicates Relative Offset support or non-support for each Information Category. For the Relative Offset supported Information Categories, the Sequence Recipient collectively indicates Continuously Increasing or Random Relative Offset requirement.

The Sequence Initiator shall follow the Relative Offset requirements of the Sequence Recipient, for Information Categories supported and not supported.

27.5 Segmentati n rul s summary

Segmentation summary rules are listed and referred to table 110.

a) The Sequence Initiator shall be responsible for segmentation. The Sequence Initiator shall follow the Relative Offset requirements of the Sequence Recipient for Information Categories.

b) An upper level at the sending end shall specify to the sending FC-2 one or more blocks to be transferred as a Sequence, the Information Category for each block, an Relative Offset space, and the Initial Relative Offset for each Information Category. The Initial Relative Offset value may be zero or non-zero.

c) The Sequence Initiator shall use the specified Relative Offset space for each Information Category and transfer one or more blocks specified in a single Sequence.

d) If the Sequence Recipient does not support Relative Offset for one or more Information Categories, the Sequence Initiator shall transmit each of these Information Categories as a contiguous block. The Sequence Initiator shall set the Relative Offset present bit in F_CTL to zero, indicating that the parameter field is not meaningful.

e) If the Sequence Recipient supports Relative Offset for one or more Information Categories and has specified during Login this support as Continuously Increasing Relative Offset, the Sequence Initiator shall transmit each of these Information Categories with Continuously Increasing Relative Offset.

   1) The Sequence Initiator shall set the Relative Offset present F_CTL bit to one.
   2) The Sequence Initiator shall use the Initial Relative Offset specified by the upper level for the Relative Offset RO_i in the first frame of the block, namely,
   $RO_i(0)$ = Initial Relative Offset (IRO_i) for the Information Category i
   3) The Sequence Initiator shall use for all other frames of the block, the Relative Offset computed as follows:
   $RO_i(n+1) = RO_i(n) +$ Length of $Payload_i(n)$
   where n is ≥ 0 (zero) and represents the consecutive frame count of frames for a

given Information Category within a single Sequence.

4) Above steps 1 through 3 shall be repeated for each block within the Sequence.

f) If the Sequence Recipient supports Relative Offset for one or more Information Categories and has specified during Login this support as Random Relative Offset, the Sequence Initiator is permitted to transmit each of these Information Categories with Random Relative Offset.

1) The Sequence Initiator shall set the Relative Offset present F_CTL bit to one.
2) The Sequence Initiator shall use the Initial Relative Offset specified by the upper level for the Relative Offset RO_i in the first frame of the subblock, namely,
$RO_i(0)$ = Initial Relative Offset (IRO_i) for the Information Category i
3) The Sequence Initiator shall use for all other frames of the subblock, the Relative Offset computed as follows:
$RO_i(n+1) = RO_i(n) + \text{Length of Payload}_i(n)$
where n is ≥ 0 (zero) and represents the consecutive frame count of frames for the subblock for a given Information Category within a single Sequence.
4) Above steps 1 through 3 shall be repeated for each subblock within the Sequence.

27.6 Reassembly rules summary

Reassembly summary rules are listed and referred to table 110.

a) The Sequence Recipient shall be responsible for reassembly of each Information Category received within the Sequence. The Sequence Recipient shall use Relative Offset or SEQ_CNT field, as specified, to perform the reassembly and make the blocks available to the receiving upper level as sent by the sending upper level.

b) The Sequence Recipient shall reassemble each Information Category within its Relative Offset space specified by the sending upper level.

c) If the Sequence R cipient has specified during Login non-support of Relative Offset for one or more Information Categories, the Sequence Recipient shall verify Relative Offset present F_CTL bit for zero value and reassemble each of these Information Categories using SEQ_CNT.

If this F_CTL bit is set to one for any of these Information Categories, the Sequence Recipient shall issue a P_RJT with a reason code of "Relative Offset not supported."

d) If the Sequence Recipient has indicated during Login Relative Offset support for one or more Information Categories and specified this support as Continuously Increasing Relative Offset, the Sequence Recipient shall verify Relative Offset present F_CTL bit for one and assemble each of these Information Categories using Relative Offset.

If the Sequence Initiator lacks the Relative Offset capability and has set the bit to zero, the Sequence Recipient shall use SEQ_CNT for reassembly.

e) If the Sequence Recipient has indicated during Login Relative Offset support for one or more Information Categories and specified this support as Random Relative Offset, the Sequence Recipient shall verify Relative Offset present F_CTL bit for one and assemble each of these Information Categories using Relative Offset.

If the Sequence Initiator lacks the Relative Offset capability and has set the bit to zero, the Sequence Recipient shall use SEQ_CNT for reassembly.

f) If the Sequence Recipient supports Continuously Increasing Relative Offset and detects random Relative Offsets, the Sequence Recipient shall issue P_RJT with the reason code of "Relative Offset out of bounds".

Table 110 - Segm ntation and r assembly rules summary

Case	Relative Offset support by Sequence Recipient	Sequence Initiator action (Segmentation)	Sequence Recipient action (Reassembly)
1	Not supported	- F_CTL Relative Offset present bit = 0 - Parameter field not meaningful	- Relative Offset shall not be used (ignore parameter field) - SEQ_CNT shall be used
		- F_CTL Relative Offset present bit = 1 - Parameter field = Relative Offset	- Issue P_RJT - Reason code = Relative Offset not supported
2	Continuously Increasing Relative Offset supported	- F_CTL Relative Offset present bit = 1 - Parameter field = Relative Offset - First frame of a block: $RO_i(0) = IRO_i$ for the block specified - All other frames of the block: $RO_i(n+1) = RO_i(n) + \text{Length of } Payload_i(n)$	Relative Offset shall be used
		- F_CTL Relative Offset present bit = 0 - Parameter field not meaningful	- Ignore parameter field - SEQ_CNT shall be used
3	Random Relative Offset supported	- F_CTL Relative Offset present bit = 1 - Parameter field = Relative Offset - Initial Relative Offset for subblocks permitted to be random - First frame of a subblock: $RO_i(0) = IRO_i$ for the subblock specified - All other frames of the subblock: $RO_i(n+1) = RO_i(n) + \text{Length of } Payload_i(n)$	Relative Offset shall be used
		- F_CTL Relative Offset present bit = 0 - Parameter field not meaningful	- Ignore parameter field - SEQ_CNT shall be used

Note:

If RO value in the Parameter field is out of bounds, the Sequence Recipient shall issue a P_RJT with a reason code of "Invalid Parameter field".

28 Connection management

The procedures for establishing and removing Class 1 Dedicated Connections are specified in this clause. See annex Q for application examples for removing a Connection.

28.1 Introduction

Class 1 Service is based on establishing a Dedicated Connection between a source N_Port and a destination N_Port. The Dedicated Connection guarantees that the full bandwidth of the Link is available to each N_Port.

Establishing a Connection

In order to establish a Class 1 Dedicated Connection, the source N_Port shall transmit a Data frame to a destination N_Port with an **SOFc1** delimiter. The Data field size of the connect-request frame is limited by the maximum buffer-to-buffer Receive Data_Field size specified by the Common Service Parameters of the Fabric (or point-to-point N_Port) or by the maximum Receive Data_Field size specified by the destination N_Port, whichever is smaller.

The N_Port shall receive an R_RDY Primitive Signal to indicate that the connect-request frame was received successfully and a buffer in the F_Port or N_Port is available. No additional frames shall be transmitted for the pending Connection until the ACK frame has been received. When the N_Port transmitting the connect-request receives an ACK (ACK_1 or ACK_N) with an **SOFn1** and with the appropriate S_ID and D_ID fields of the connect-request frame, a Dedicated Connection is established.

NOTES

1 A Connection is established from the N_Port's perspective (Connection Initiator) at the time the N_Port receives the ACK frame. It has no relation to the method or timing by which the Fabric actually forms the Dedicated Connection.

2 It is recommended that a Fabric not introduce a phase discontinuity (see 5.3) while establishing a Class 1 Connection. No frame errors are introduced by such a discontinuity.

When a Dedicated Conn ction is established:

— Both N_Ports shall reinitialize their end-to-end Class 1 Credit to the Login values.

— the N_Port initiating the connection shall be known as the Connection Initiator and the N_Port responding to the connect-request shall be known as the Connection Recipient.

— the Connection Initiator may continue the initial Sequence, and

— the Connection Recipient may initiate new Sequences with an **SOFi1** delimiter when a Dedicated Connection is not unidirectional, as indicated by the setting of F_CTL bit 8 (see 28.5.3).

Removing a Connection

Removing a Dedicated Connection is accomplished by either N_Port transmitting a frame terminated by an **EOFdt** or the transmission or reception of the Link Reset Primitive Sequence. Normally, removing a Dedicated Connection shall be negotiated between the two N_Ports involved. Negotiation is required in order to avoid breaking the Dedicated Connection while frames are still flowing between the N_Ports.

In urgent situations which do not require the use of the Link Reset Primitive Sequence, an N_Port may request the immediate removal of a Dedicated Connection by transmitting the Remove Connection Basic Link Service frame. The ACK frame transmitted in response shall end with an **EOFdt** delimiter.

The Fabric terminates a Dedicated Connection after an **EOFdt** or **EOFdti** passes through the F_Port in either direction. Any frames flowing in either direction at the time the Connection is removed may be corrupted.

End_Connection (E_C) is the control bit in F_CTL which is used to perform the negotiation.

Frame processing precedence

There are a number of different fields within a frame which affect the processing for establishing and removing Dedicated Connections. The general precedence hierarchy follows:

a) Frame delimiters (**SOFc1**, **EOFdt**),

b) R_CTL

– Data frame, or
– Link_Control frame.

c) F_CTL

– Unidirectional Connection, when meaningful (bit 8),
– End_Sequence (bit 19),
– other bits when meaningful (for example, if bit 19=1, check End_Connection, Chained_Sequence, and so forth) as in tables 38 and 39.

28.2 Applicability

Connection management applies to Class 1 Service. An N_Port supporting Class 1 Service may also support Class 2 and Class 3. Depending on the options supported by the Fabric, multiple class support by an N_Port may be complex. Because Class 1 involves Dedicated Connections, managing Class 1 usually overrides Class 2 or 3 management. FC-PH specifies the allowable responses on a Class by Class basis.

NOTE - An N_Port engaged in Class 2 communication with another N_Port may experience Sequence timeouts if the other N_Port supports Class 1 and Class 2 without functioning in Intermix mode and a Class 1 Connection is created to the other N_Port.

28.3 Topology models

An N_Port may be attached directly to another N_Port through a point to point connection or through a Fabric. The topology may be determined using the **explicit** Login procedure. Topology may also be determined by a method not defined by FC-PH.

28.3.1 Fabric model

The N_Port behavior described in this clause is based on a Fabric model in which an F_Port acts as the control point for establishing and removing Class 1 connections on behalf of the locally attached N_Port. The N_Port relies on specific behavioral characteristics in order to base its operation.

The following terminology is used in the discussion of Connection management. N_Port (A), (B), or (X) describe three separate N_Port Identifiers where A ≠ B ≠ X. The side of the F_Port directly attached to the N_Port side is termed its "Link" side, whereas the side of the F_Port attached internally to the Fabric is termed its "internal" side.

The following F_Port characteristics are required behavior:

a) When an F_Port is not connected, it may receive a connect-request and begin processing that request. The process of acting on that request is termed accepting the connect-request.

b) After an F_Port has accepted a connect-request from the Link, it reserves Fabric resources as it attempts to establish the requested Dedicated Connection.

   – the F_Port is Busy to other connect-requests from its internal side as destination N_Port.

   – the F_Port returns an F_BSY with **EOFdt** to the Link if a busy condition is encountered. (If a Fabric supports stacked connect-requests, the period of time before issuing an F_BSY may be extended.)

   – the F_Port returns an F_RJT with **EOFdt** to the Link if a reject condition is satisfied.

c) After an F_Port has accepted a connect-request from the internal side:

   – it passes the connect-request to the Link as the destination N_Port.

   – it monitors its Link side for a proper confirmation response frame expected in response to the delivered connect-request.

   – it discards connect-request frames (**SOFc1**) received from its Link side. (If the Fabric supports stacked connect-requests, the connect-request received from its Link side shall be stacked for establishing a Dedicated Connection at a later time.)

d) If an F_Port encounters a collision case wherein connect-requests from both the internal side and the Link side arrive simultaneously, the F_Port accepts the connect-request from the internal side and proceeds as in step c above.

e) If a failure is detected within a Fabric such that a Dedicated Connection can no longer be used, the Fabric may notify each of the F_Ports attached to the N_Ports involved in an established or pending Dedicated Connection and each of the F_Ports may then transmit the

Link Reset Primitive Sequence to its locally attached N_Port (i.e., initiate Link Reset Protocol).

f) See 16.4 for information on the effect of Primitive Sequences on F_Ports.

28.3.2 Point to Point model

A point to point topology is indicated during the Login procedure. Two N_Ports arranged in a point to point connection may choose to:

- establish one Dedicated Connection for the duration of an operating period, or
- establish and remove Dedicated Connections dynamically, as the need to communicate arises.

28.4 Connect / disconnect rules

28.4.1 Connect-request rules

The following sections specify the rules governing the behavior of the source and destination of the connect-request.

28.4.1.1 Source of connect-request

The following rules specify the connect-request procedure as the source (A) of the connect-request:

a) An N_Port (A) shall initiate a connect-request using a Data (Device_Data or Link_Data) frame with an **SOFc1** delimiter directed to destination N_Port (B). The connect-request frame is formed as follows:

- an **SOFc1** delimiter
- a Data (Device_data or Link_Data) frame
- an S_ID of (A) and a D_ID of (B)
- the E_C bit (F_CTL bit 18) shall be set = 0
- the Unidirectional bit (F_CTL bit 8) shall be set = 0 for a bidirectional Connection, and = 1 for a unidirectional Connection
- an **EOFn** ending delimiter

b) The Data Field of the connect-request shall be limited to the smaller of

- the maximum buffer-to-buffer Receive_Data_Field size specified by the Fabric, if present, or
- the maximum Receive_Data_Field size specified by the destination N_Port.

c) After N_Port (A) transmits the connect-request frame, N_Port (A) shall wait for a response frame before transmitting another frame for this Sequence. Additional Sequences for this Connection shall not be initiated until the Dedicated Connection is established.

d) A Dedicated Connection is established when the connect-request frame has been responded to by an ACK frame. A proper response frame consists of:

- an ACK_1 or ACK_N frame with
  - an **SOFn1** delimiter, and
  - an S_ID of (B), and a D_ID of (A)
  - an **EOFn**, or **EOFt** delimiter

e) An alternate response frame is also possible from the destination N_Port:

- a P_BSY or P_RJT frame with
  - an **SOFn1** delimiter,
  - an S_ID of (B), and a D_ID of (A), and
  - an **EOFdt** delimiter.

f) An alternate response frame is also possible from the Fabric:

- an F_BSY or F_RJT frame with
  - an **SOFn1** delimiter,
  - an S_ID of (B), and a D_ID of (A), and
  - an **EOFdt** ending delimiter.

g) After a Dedicated Connection is established, N_Port (A) shall be the Connection Initiator and N_Port (B) shall be the Connection Recipient.

h) After a Dedicated Connection is established (i.e., the ACK to the connect-request has been received), the Connection Initiator, N_Port (A), may continue transmitting its initial Sequence and initiate other Sequences with **SOFi1** up to N_Port (B)'s ability to support Concurrent Sequences.

28.4.1.2 Destination of connect-request

The following rules specify the connect-request procedure at the destination (B) of the connect-request:

a) If a Data frame started by **SOFc1** is received when N_Port (B) is not connected and N_Port (B) is busy, N_Port (B) responds with P_BSY with an **EOFdt** delimiter as specified in 28.4.1.1 item number e.

b) If a Data frame started by **SOFc1** is received when N_Port (B) is not connected and N_Port

(B) rejects the connect-request, N_Port (B) responds with P_RJT with an **EOFdt** delimiter as specified in 28.4.1.1 item number e.

c) If a Data frame started by **SOFc1** is received when N_Port (B) is not connected and not busy, N_Port (B) responds with the proper response frame. A proper response frame is defined in 28.4.1.1 item number d.

A Dedicated Connection is established with N_Port (A). N_Port (B) shall be the Connection Recipient and N_Port (A) shall be the Connection Initiator.

d) With a Fabric present, if N_Port (B) receives a connect-request frame from N_Port (X) after a connect-request has been transmitted by N_Port (B), N_Port (B) requeues its own request for transmission at a later time and responds with a proper response frame to N_Port (X).

NOTE - N_Port (B) requeues its original connect-request because the Fabric has discarded it. N_Port (B) needs to adjust its end-to-end Credit_CNT to account for the discarded connect-request.

If stacked connect-requests are being employed, the connect-request shall not be requeued by N_Port (B).

A Dedicated Connection is established with N_Port (X) with N_Port (B) as Connection Recipient.

e) Without a Fabric present, if N_Port (B) accepts a connect-request frame from N_Port (A) after a connect-request has been transmitted by N_Port (B), N_Port (B) shall respond as follows:

1) if A > B, in value,

— N_Port (B) requeues its own request for transmission at a later time,

— responds to (A) with a proper response frame,

— a Dedicated Connection is established with N_Port (A) with N_Port (B) as Connection Recipient, and

— N_Port (B) may reinitiate its connect-request Sequence using **SOFi1**.

2) if A < B, in value,

— N_Port (B) discards connect-request from (A), and

— waits for a proper response frame.

f) After a Dedicated Connection is established (i.e., the ACK to the connect-request is transmitted), N_Port (B) may begin initiating Sequences with **SOFi1** up to the destination N_Port's ability to support Concurrent Sequences when the Connection is bidirectional.

28.4.2 Connection Rules

a) If a bidirectional Connection is established (F_CTL bit 8 = 0 on connect-request), it shall remain bidirectional for the life of the Connection.

b) If a unidirectional Connection is established (F_CTL bit 8 = 1), it shall remain unidirectional until the Connection Initiator sets F_CTL bit 8 = 0, if ever, on the first or last frame of a Sequence (see 18.5) subsequent to the connect-request frame.

c) If a unidirectional Connection is made bidirectional, it shall remain a bidirectional Connection for the life of the Connection.

d) The Connection Recipient may request that a unidirectional Connection be made bidirectional by setting F_CTL bit 8 = 0 on an ACK in response to a Data frame. Once F_CTL bit 8 is set to 0 on an ACK, it shall remain set to 0 for the life of the Connection.

28.4.3 Remove Connection rules

a) An Active Sequence is complete when the corresponding ACK response to the last Data frame has been transmitted by the Sequence Recipient from the Recipient's perspective and has been received by the Sequence Initiator from the Initiator's perspective.

b) If an N_Port detects an abnormal condition which requires immediate removal of an existing Connection, the N_Port shall use the Remove Connection (RMC) Basic Link Service frame with the appropriate F_CTL bit settings which includes setting the E_C bit = 1 in order to remove the Dedicated Connection. All Open and Active Class 1 Sequences are abnormally terminated and left in an indeterminate stat relative to the Upper Level Protocol. RMC shall not b used in place of Link Reset in protocols which require Link Reset.

c) An N_Port shall transmit the E_C bit in F_CTL set to one on the last Data frame of a Sequence to indicate:

   – it is ready to end the Connection,
   – it shall not initiate any new Sequences, and
   – it requests the other N_Port to complete its Active Sequences and not initiate any new Sequences.

d) If an N_Port has transmitted the E_C bit in F_CTL set to one and it receives a Data frame initiating a new Sequence, it shall respond as though the Sequence had been initiated before the E_C bit had been transmitted as one.

e) If an N_Port has transmitted the E_C bit in F_CTL set to one and it receives the last Data frame of a Sequence with the Chained_Sequence (C_S) bit in F_CTL set to one, it shall respond as though it had not previously transmitted the E_C bit (it shall set the E_C bit on a future Sequence).

f) If either the Connection Initiator or Connection Recipient has completed its last Active Sequence of the existing Connection and it receives a Data frame with E_C set to one, it shall transmit the corresponding ACK with an **EOFdt** delimiter.

g) If either the Connection Initiator or Connection Recipient receives a Data frame with the E_C bit set to one and it has not completed all of its Active Sequences, it

   – does not initiate any new Sequences (unless C_S bit is received),
   – completes Active Sequences,
   – transmits the E_C bit set to one on the last Data frame of its last Active Sequence for the Connection.

h) If an N_Port encounters a collision case wherein a Data frame has been transmitted with E_C set to one and a Data frame is received with E_C set to one before receiving its ACK,

   – the Connection Recipient shall respond with an ACK with an **EOFt** delimiter, whereas
   – the Connection Initiator shall withhold transmitting ACK until after its Sequence is complete.
   – the Connection Initiator shall transmit the ACK with the **EOFdt** delimiter.

28.5 Establishing a Connection

A Dedicated Connection is established with an N_Port as the source of a connect-request (Connection Initiator) or as the destination N_Port (Connection Recipient) of a connect-request from another Class 1 N_Port.

28.5.1 Connection Initiator

When FC-2 receives a request from an FC-4 or upper level to initiate a Class 1 Sequence when a Dedicated Connection does not exist, the N_Port shall also establish a Class 1 Connection with the destination N_Port as part of the Sequence initiation.

The N_Port (A) initiates the connect-request using a Data (Device_Data or Link_Data) frame with an **SOFc1** delimiter. The Data Field size is limited to the smaller of:

a) the maximum buffer-to-buffer Receive_Data_Field size specified by the Fabric, if present, or
b) the maximum Receive_Data_Field size specified by the destination N_Port.

After the N_Port transmits the connect-request frame, no additional frames shall be transmitted for any Sequence for the pending Connection until a response frame has been received. The N_Port receives an R_RDY Primitive in response to the connect-request to indicate successful delivery to the F_Port or N_Port and that a buffer is available for a connectionless frame. If an N_Port is not operating in Intermix mode, the N_Port shall not transmit Class 2 or 3 frames until the pending Dedicated Connection is removed. A Dedicated Connection is not established until a proper ACK frame is received from the destination N_Port. A proper ACK frame is defined in 28.4.1.1 item number d.

Table 111 defines Event 1 as the connect-request and events 2 through 9 define the possible responses.

Table 111 - Responses to c nnect-request (SOF$_{c1}$)						
Event	SOF	D_ID	S_ID	Frame	EOF	N_Port Action
1.	SOF$_{c1}$	B	A	Data frame	EOF$_n$	-Transmit connect-request -Wait for confirmation frame
2.	SOF$_{n1}$	A	B	F_BSY	EOF$_{dt}$	Connection failed, Busy in Fabric
3.	SOF$_{n1}$	A	B	P_BSY	EOF$_{dt}$	Connection failed, Busy in N_Port
4.	SOF$_{n1}$	A	B	F_RJT	EOF$_{dt}$	Connection failed, Fabric Reject
5.	SOF$_{n1}$	A	B	P_RJT	EOF$_{dt}$	Connection failed, Port Reject
6.	SOF$_{n1}$	A	B	ACK_1 or ACK_N	EOF$_n$ EOF$_t$	-Dedicated Connection established -Continue transmitting Sequence -Sequence ended, Connection established
7.	SOF$_{c1}$	A	B	Data frame	EOF$_n$	N_Port A responds as follows PTP: If A > B in value, -discard B's frame and wait for ACK response. -Dedicated Connection established with B. PTP: If A < B in value, -respond with SOF$_{n1}$ on ACK -Dedicated Connection established with B. -retransmit request assoc with event 1 with SOF$_i$
8.	SOF$_{c1}$	A	X	Data frame	EOF$_n$	Fabric is present. -Requeue request assoc with event 1 (unless stacked connect-requests used) -Respond with SOF$_{n1}$ on ACK_1 or ACK_N. -Dedicated Connection established with X.
9.						-Timeout, no response frame. -Perform Link Reset Protocol. (see 16.6.5)

Event 1

A connect-request is transmitted by N_Port (A) with an **SOF$_{c1}$** delimiter with a destination of N_Port (B).

Event 2

An F_BSY indicates that the Fabric is unable to access the destination N_Port due to a busy condition internal to the Fabric. Try again later.

Event 3

A P_BSY indicates that the destination N_Port link facility is temporarily occupied with other activity and unable to accept the connect-request. Try again later.

Event 4

An F_RJT indicates that the Fabric is unable to establish the Dedicated Connection. The reason code specifies the cause.

Event 5

A P_RJT indicates that the destination N_Port is unable to establish the Dedicated Connection. The reason code specifies the cause.

Event 6

A Dedicated Connection has been established. N_Port (A) is Connected to N_Port (B).

a) N_Port (A) is the Connection Initiator and N_Port (B) is the Connection Recipient.

b) N_Port (A) may continue transmitting the Sequence initiated (**EOF$_n$**), or the Sequence which initiated the Connection may be complete (**EOF$_t$**).

c) N_Port (A) may initiate other Sequences with the same destination N_Port (B) up to the maximum number of Sequences defined by the Service Parameters obtained from (B) during Login.

d) The connected N_Port (B) may initiate Sequences using **SOF$_{i1}$** to start each Sequence when the Connection is

bidirectional. The number of Active Sequences is limited by the Service Parameters provided by N_Port (A) during Login.

Event 7

In the case of a point to point (PTP) topology, if (A) is greater than (>) (B) in absolute value, then N_Port (A) discards the connect-request and waits for a response (N_Port (B) requeues request with **SOFi1**).

In the case of a PTP topology where (A) < (B), N_Port (A) terminates its own previous connect-request with the intent of retransmission at a later time (i.e., requeues Event 1 with **SOFi1**). N_Port (A) responds as the destination of the connect-request with an appropriate ACK frame and becomes the Connection Recipient. N_Port (B) is the Connection Initiator.

Event 8

In the case of a Fabric present, N_Port (A) terminates its own previous connect-request with the intent of retransmission at a later time (i.e., requeues Event 1 with **SOFi1** or **SOFc1**, as appropriate). N_Port (A) responds as the destination of the connect-request with an appropriate ACK frame and becomes the Connection Recipient. N_Port (X) is the Connection Initiator. If stacked connect-requests are functional, then it is unnecessary to requeue its own previous connect-request.

Event 9

If a frame response is not received within the timeout period (E_D_TOV), a Link timeout shall be detected and the Link Reset Protocol shall be performed (see 16.6.5).

See 28.4.1 for the rules which a source N_Port of a connect-request shall follow.

28.5.2 Stacked connect-requests

Stacked connect-requests is a feature which may be provided by a Fabric. Support for Stacked connect-requests is determined by an N_Port during Login with the Fabric. Intermix shall also be functional in order to use this feature. If Stacked connect-requests are functional, an N_Port may transmit one or more connect-requests (**SOFc1**) without regard as to whether a Connection is pending, established, or complete.

The Fabric may process multiple connect-requests in any order and shall provide the status of the Stacked request via the Read Connection Status Extended Link Service command. Stack d connect-requests allow the Fabric to work on establishing a Connection while an N_Port is busy servicing another Connection and may enhance system performance. An N_Port uses the E_D_TOV timeout period after a connect-request has been transmitted (part of Sequence timeout) whether the connect-request is a normal request or a Stacked request. That is, an ACK response shall be received within an E_D_TOV timeout period or an F_BSY shall be returned from the Fabric to the N_Port. If either condition is not met within E_D_TOV, the N_Port detects a Sequence timeout and Connection Recovery (see 28.8) is performed. A Fabric which supports stacked connect-requests shall deal with any associated race conditions which occur during the establishment and removal of connections.

Due to the timing relationships involved, there are two methods of implementing stacked connect-requests by a Fabric:

a) transparent mode, or
b) lock-down mode.

In transparent mode, when the **SOFc1** Data frame is delivered to the destination N_Port, the return path of the bi-directional circuit is established in the same manner as Exclusive Dedicated Connections. This means that the destination N_Port of the **SOFc1** is able to transmit Data frames immediately following transmission of the ACK frame in response to the **SOFc1** frame.

In lock-down mode, when the **SOFc1** Data frame is delivered to the destination N_Port, the return path of the bi-directional circuit is not necessarily established to the source N_Port of the **SOFc1**. Therefore, the **SOFc1** Data frame shall also set F_CTL bit 8 = 1 (Unidirectional Transmit) in order to inhibit the destination N_Port of the **SOFc1** from sending any Data frames after the ACK frame is transmitted in response to the connect-request.

The determination of what mode of connect-request is functional is based on the FLOGI request and Accept reply from Fabric Login. The information is contained in Class Service Parameters Word 0, Bits 29-28.

Bit 29 is a request by the N_Port for transparent stacked connect-requ sts. Bit 28 is a request by the N_Port for lock-down stack d connect-

requests. An N_Port may request either one or both modes. However, a Fabric shall support either transparent or lock-down or none. The Fabric shall not support both modes.

Word 0, bit 29 Transparent mode - stacked connect-request

0 = Transparent mode not requested
1 = Transparent mode requested

When an N_Port performs Login with a Fabric, it shall request support for transparent stacked connect-requests by specifying bit 29 = 1. Bit 29 = 1 is only meaningful if Intermix has also been requested. If the Accept reply from the Fabric specifies bit 29 = 1, then both the N_Port and Fabric have agreed that transparent stacked connect-requests are functional.

The following set of values specifies the meaning of the combination of Word 0, bit 29:

N_Port	F_Port	
0	0	Neither supports
0	1	Fabric is capable of supporting, transparent mode stacked connect-requests not functional
1	0	N_Port support requested, Fabric does not support
1	1	N_Port requested, Fabric agrees, transparent mode stacked connect-requests are functional

Word 0, bit 28 Lock-down mode - stacked connect-request

0 = Lock-down mode not requested
1 = Lock-down mode requested

When an N_Port performs Login with a Fabric, it shall request support for lock-down stacked connect-requests by specifying bit 28 = 1. Bit 28 = 1 is only meaningful if Intermix has also been requested. If the Accept reply from the Fabric specifies bit 28 = 1, then both the N_Port and Fabric have agreed that lock-down stacked connect-requests are functional.

The following set of values specifies the meaning of the combination of Word 0, bit 28:

N_Port	F_Port	
0	0	Neither supports
0	1	Fabric is capable of supporting, lock-down mode stacked connect-requests not functional
1	0	N_Port support requested, Fabric does not support
1	1	N_Port requested, Fabric agrees transparent mode stacked connect-requests are functional

28.5.3 Unidirectional Dedicated Connection

When an N_Port transmits a connect-request, it may set F_CTL bit 8 = 1 to indicate that a unidirectional Dedicated Connection is being established. Unidirectional means that only the N_Port transmitting the connect-request (Connection Initiator) shall be able to transmit Data frames. The destination of the connect-request (Connnection Recipient) shall only transmit ACK or Link Response frames (P_BSY, P_RJT).

F_CTL bit 8 is meaningful on the first Data frame of a Sequence and the last Data frame of a Sequence. The Connection Initiator may reset F_CTL bit 8 = 0 at the end of the first Sequence (if multi-frame), or at the start or end of any subsequent Sequence. If F_CTL bit 8 is reset to = 0 on a meaningful Data frame (first or last) of a subsequent Sequence, the Connection becomes bidirectional for the remainder of the Connection.

If the Connection Recipient of a unidirectional Connection desires that the Connection become bidirectional, it may request that change by setting F_CTL bit 8 to 0 in the ACK response to any Data frame in a Sequence. Once the Connection Recipient has transmitted F_CTL bit 8 = 0 on an ACK, it shall continue to do so for the remainder of the Connection (if set to 1, it shall be ignored). The request for a bidirectional Connection may not be honored by the Connection Initiator.

The unidirectional transmit bit (F_CTL bit 8) has three primary uses:

a) lock-down stacked connect-requests,
b) Connection Initiator with temporary receive buffer allocation problems (i.e., is unable to receive as much data as th end-to-end Credit promised at Login du to buffering for transmit), or

c) an implementation that may only receive or transmit Data frames at any instant in time.

28.5.4 Destination of connect-request

When N_Port (B) is not connected, but is available, and it receives a connect-request as the destination N_Port, N_Port (B) transmits the appropriate ACK frame (ACK_1 or ACK_N) to N_Port (A) which is requesting the connection. After the ACK frame has been transmitted with **SOFn1**, a Dedicated Connection is established with N_Port (A) as the Connection Initiator and N_Port (B) as the Connection Recipient. After a Connection has been established, N_Port (B) may initiate Sequences with the N_Port (A) using an **SOFi1** delimiter.

If N_Port (B) is not connected, but is busy, and it receives a connect-request as the destination N_Port from N_Port (A), it responds with P_BSY with an **EOFdt** delimiter.

See 28.4.1 for the rules which a destination N_Port of a connect-request shall follow.

28.6 Connected

When an N_Port is in a Dedicated Connection, it may receive Sequences as the Sequence Recipient as well as initiate Sequences as the Sequence Initiator.

When an N_Port has acted as a Connection Initiator and is in a Unidirectional Connection, it may only initiate Sequences and shall not receive Sequences. When an N_Port has been the Connection Recipient and is in a Unidirectional Connection, it may only receive Sequences and shall not initiate Sequences.

28.7 Removing a Connection

Class 1 Connections may be removed in four ways:

- using the remove connection procedure which uses the E_C bit in F_CTL for negotiation,
- the Remove Connection command, which forces the recipient N_Port to transmit **EOFdt**,
- the Link Reset Protocol, or
- transmission or reception of any Primitive Sequence.

The first two methods use the **EOFdt** delimiter to remove the Connection. The third method uses the Link Reset Primitive Sequence to remove the Connection.

28.7.1 When to remove a Connection

FC-PH does not specify the method to be employed in determining when to end a connection. Normally, an N_Port chooses to remove a Dedicated Connection when it has no Sequences to transmit to the Connected N_Port and it has a request to transmit Sequences to a different N_Port. In addition, F_CTL bits offer one alternative to suggest that removal might be appropriate (Continue Sequence condition bits) and one alternative that restricts an N_Port from removing a Connection while using the E_C bit in the normal remove connection procedure (Chained_Sequence).

NOTE - In order to maximize system utilization, an N_Port should remove a Dedicated Connection when it has no Data to transmit in order to be available for Connection to another N_Port.

Continue Sequence condition bits
Two information bits are available in F_CTL to indicate when the next Sequence of an Exchange is estimated to be transmitted: immediately (0 1), soon (1 0), or delayed (1 1). The indication of delayed transmission may be used as one condition to assist in determining the appropriate time to remove a Dedicated Connection.

Chained_Sequence (C_S)
The C_S bit in F_CTL shall be used in the last Data frame of a Sequence in conjunction with Sequence Initiative transfer to indicate that the transmitter of the C_S = 1 requires a reply Sequence before the existing Dedicated Connection is removed.

28.7.2 End_Connection bit

An E_C bit in F_CTL is used during the remove connection procedure. By monitoring both transmission of the E_C bit as well as reception of the E_C bit, each N_Port is able to determine the appropriate circumstanc s to transmit an **EOFdt** in order to remove the connection. The E_C bit in F_CTL shall be set to zero on a conn ct-request frame (**SOFc1**) in order to avoid ambig-

uous error scenarios where the ACK (**EOFdt**) is not properly returned to the Connection Initiator.

E_C shall be transmitted as zero in a frame to indicate:

- This N_Port wishes to maintain the existing Dedicated Connection.
- This N_Port may transmit another Sequence within this Connection.
- This N_Port may wait for a reply Sequence within this Connection.

E_C shall be transmitted as one on a Data frame to indicate:

- This N_Port is ready to remove the existing Dedicated Connection.
- This N_Port has completed all Active Sequences and agrees not to initiate another Sequence unless it receives a Chained_Sequence indication on the last Data frame of a Sequence being normally terminated.
- This N_Port is requesting the destination N_Port to complete Active Sequences in progress and not to initiate any new Sequences.

E_C shall be transmitted set to one on the last Data frame of the last Active Sequence for this Connection to indicate that this N_Port requests the receiving N_Port to transmit **EOFdt** on the ACK response frame if the receiving N_Port has completed all Active Sequences, otherwise, transmit an ACK frame with **EOFt**. When the N_Port which received the E_C bit set to one is completing its last Sequence, it shall transmit the E_C bit set to one on the last Data frame of its last Sequence.

28.7.3 EOFdt transmission

EOFdt is transmitted by either the Connection Initiator or the Connection Recipient on an ACK frame corresponding to the last Data frame of a Sequence with the E_C bit set to one if the N_Port receiving the Data frame has completed its last Active Sequence of the existing Connection.

If both N_Ports have transmitted their last Data frame of a Sequence with E_C set to one but have not received the ACK frame to complete the Sequence, the Connection Initiator delays transmitting ACK until it has received the ACK for its last Sequence. After the Connection Initiator has completed its last Sequence, it transmits the final ACK to the Connection Recipient with the **EOFdt** delimiter to remove the Connection.

The **EOFdt** is also used on the ACK responding to the RMC Basic Link Service command in order to immediately remove an existing Connection.

28.8 Connection Recovery

In case of link errors, the state of the existing Dedicated Connection may not be known with certainty. When the state of the Dedicated Connection is unknown, the Link Reset Protocol shall be used to remove the Connection and establish a known state (see 16.6.5). Errors within a specific Sequence are processed according to the rules for handling Sequence errors.

For example, the following events provide instances when an N_Port is unable to determine the state of an existing Dedicated Connection and should perform the Link Reset Protocol. N_Port (A) believes that a Dedicated Connection exists with N_Port (B):

- N_Port (A) receives an **SOFc1** from N_Port (C),
- N_Port (A) receives an **SOFi1** from N_Port (C), or
- N_Port (A) receives an **SOFn1** from N_Port (C).

28.8.1 Link timeout

When the last Active Sequence during a Class 1 Dedicated Connection has detected a Sequence timeout, a Link Timeout is detected. For more discussion on Link Timeout see 29.2.3. The Link Reset Protocol is performed as d scribed in 16.6.5.

28.8.2 Corrupted connect-requ st

If an N_Port is not engaged in a Class 1 Dedicated Connection, and it receives a frame started by **SOF_{c1}** and the frame is detected as invalid, the N_Port discards the frame and performs the Link Reset Protocol.

29 Error detection/recovery

29.1 Introduction

Link integrity and Sequence integrity are the two fundamental levels of error detection in FC-PH. Link integrity focuses on the inherent quality of the received transmission signal. When the integrity of the link is in question, a hierarchy of Primitive Sequences are used to reestablish link integrity. When Primitive Sequence Protocols are performed, additional data recovery on a Sequence basis may be required.

A Sequence within an Exchange provides the basis for ensuring the integrity of the block of data transmitted and received. Exchange management ensures that Sequences are delivered in the manner specified by the Exchange Error Policy (see 29.6.1.1). Each frame within a Sequence is tracked on the basis of Exchange_ID, Sequence_ID, and a sequence count within the Sequence. Each frame is verified for validity during reception. Sequence retransmission may be used to recover from any frame errors within the Sequence and requires involvement, guidance, or authorization from an upper level.

Credit loss is an indirect result of frame loss or errors. Credit loss is discussed in regard to methods available to reclaim apparent lost Credit resulting from other errors. See clause 26 for a more complete discussion on flow control, buffer-to-buffer Credit, and end-to-end Credit.

29.2 Timeouts

29.2.1 Timeout periods

Three timeout values are specified in FC-PH. The actual value used for a timeout shall not be less than the specified value and shall not exceed the specified value by more than 20%.

29.2.1.1 R_T_TOV

The Receiver_Transmitter timeout value (R_T_TOV) is used by the receiver logic to detect Loss of Synchronization (see 12.1.1.2). The value for R_T_TOV is 100 ms.

29.2.1.2 E_D_TOV

A short timeout value is known as the Error_Detect_Timeout Value (E_D_TOV). The E_D_TOV is used as the timeout value for detecting an error condition. The value of E_D_TOV represents a reasonable timeout value for detection of a response to a timed event. For example, during Data frame transmission it represents a timeout value for a Data frame to be delivered, the destination N_Port to transmit a Link_Response, and the Link_Response to be delivered to the Sequence Initiator. The E_D_TOV value selected should consider configuration and N_Port processing parameters. A default value is 10 s. However, a valid E_D_TOV value shall also adhere to the proper relationship to the R_A_TOV value. When an N_Port performs Fabric Login, the Common Service Parameters provided by the F_Port specify the proper value for E_D_TOV.

When an N_Port performs N_Port Login in a point-to-point topology, the Common Service Parameters provided by each N_Port specify a value for E_D_TOV. If the two values differ, the longer time shall be used by each N_Port. An N_Port may determine another N_Port's value for E_D_TOV via the Read Timeout Value (RTV) Link Service command (see 21.4.13). Timeout values as specified in the Payload of the Accept are counts of 1 ms increments. Therefore, an E_D_TOV value of hex '0000000A' specifies a time period of 10 ms.

There are three cases when E_D_TOV is used as an upper limit, that is, an action shall be performed in less than an E_D_TOV timeout period:

- transmission of consecutive Data frames within a single Sequence,
- retransmission of a Class 2 Data frame in response to an F_BSY or P_BSY, and
- transmission of ACK frames from the point in time that the event which prompted the acknowledgment action.

For all other cases, E_D_TOV shall expire before an action is taken. Two such examples include:

- Link timeout (see 29.2.3), and
- Sequence timeout (see 29.2.4).

29.2.1.3 R_A_TOV

A long timeout value is known as the Resource_Allocation_Timeout Value (R_A_TOV). The R_A_TOV is used as the timeout value for determining when to Reinstate a Recovery_Qualifier. The value of R_A_TOV represents E_D_TOV plus twice the maximum time that a frame may be delayed within a Fabric and still be delivered. The default value of R_A_TOV is 120 s.

When an N_Port performs Fabric Login, the Common Service Parameters provided by the F_Port specify the proper value for R_A_TOV. When an N_Port performs N_Port Login in a point-to-point topology, the Common Service Parameters provided by each N_Port only specify a value for E_D_TOV. R_A_TOV shall be set to twice the E_D_TOV value in a point-to-point topology. An N_Port may determine another N_Port's value for R_A_TOV via the Read Timeout Value (RTV) Link Service command (see 21.4.13).

When R_A_TOV is used to determine when to reuse an N_Port resource such as a Recovery_Qualifier, the resource shall not be reused until R_A_TOV has expired for all frames previously transmitted which fall within the SEQ_CNT range of the Recovery_Qualifier. This may be accomplished by restarting the R_A_TOV timer as each Data frame of a Sequence is transmitted. Other techniques may also be employed in order to accomplish equivalent behavior.

From the Fabric viewpoint, the maximum time that a frame may be delayed within the Fabric and still be delivered is in the range of 1 to 2^{31} -1 ms. The Fabric shall ensure delivery within the maximum delivery time by requiring each Fabric Element to timeout frames stored in receive buffers within the Element. Individual Elements may use different timeout values, possibly one for each service class. The maximum Fabric delivery time is variable and is the cumulative timeout value for elements along the path or paths joining the source and destination N_Ports.

29.2.2 Link Failure tim uts

A Link Failure is detected under the following timeout conditions:

- Loss of Signal (see 12.1.1.2),
- Loss of Synchronization > timeout period (R_T_TOV) (see 12.1.1.2),
- Link Reset Protocol timeout (> R_T_TOV) (see 16.6.5).

29.2.3 Link timeout

In Class 1, a Link timeout error is detected during a Dedicated Connection if Sequence timeouts have been detected for all Active Sequences.

In Class 1 (**SOFc1**), Class 2, or Class 3, a Link timeout error shall be detected if one or more R_RDY Primitive Signals are not received within the timeout period (E_D_TOV) after the buffer-to-buffer Credit_CNT has reached zero.

Recovery from Link timeout is accomplished by performing the Link Reset Protocol (see 16.6.5).

Link timeout values need to take Fabric characteristics into consideration. In Class 1, the concern is the time required by the Fabric to process a connect-request. In Class 2 and 3, the concern is the maximum time required for frame delivery by the worst case route with any associated delays within the Fabric.

29.2.4 Sequence timeout

The basic mechanism for detecting errors within a Sequence is the Sequence timeout. Other mechanisms which detect frame errors within a Sequence are performance enhancements in order to detect an error sooner than the timeout period. Since an Active Sequence utilizes N_Port resources, Sequence timeout is applicable to both discard and process error policies. The Sequence timeout mechanism varies by class.

29.2.4.1 Class 1 and 2

Both the Sequence Initiator and the Sequence Recipient use a timer facility with a timeout period (E_D_TOV) between expected events. The expected event for the Initiator to Data frame transmission is an ACK response (or alternate). The expected event for the Recipient is another Data frame for the same Sequence which is Active and incomplete. Other events such as Link Credit Reset and Abort Exchange shall also stop the Sequence timer. When a Sequence Recipient receives the last Data frame transmitted for the Sequence, it shall verify that all frames have been received before transmitting the final ACK (**EOFt** or **EOFdt**) for the Sequence.

If the timeout period (E_D_TOV) expires for an expected event before the Sequence is complete, a Sequence timeout is detected. Timeouts are detectable by both the Sequence Initiator and the Sequence Recipient. If a Sequence Initiator detects a Sequence timeout, it shall transmit the ABTS frame to perform the Abort Sequence Protocol. If a timeout is detected by the Sequence Initiator before the last Data frame is transmitted, ABTS notifies the Sequence Recipient that an error was detected by the Sequence Initiator (see 29.7.1.1). Detection of a Sequence timeout by the Sequence Initiator may also result in aborting the Exchange (see 21.2.2 and 21.4.2).

If a Sequence Recipient detects a Sequence timeout, it shall set the Abort Sequence Condition (F_CTL bits 5-4) in an ACK to an 0 1, or 1 1 value to indicate a missing frame error condition. The Sequence Recipient shall also post the detected condition in the Exchange Status Block associated with the Sequence. A Sequence timeout results in either aborting the Sequence (see 21.2.2) by the Sequence Initiator, abnormal termination of a Sequence (see 29.7.1) by the Sequence Recipient, or aborting the Exchange by either the Sequence Initiator or Sequence Recipient (see 21.4.2).

In Class 2, if a Sequence has been aborted and the Sequence Recipient supplies the Recovery_Qualifier (OX_ID, RX_ID, and a SEQ_CNT range, low and high SEQ_CNT values), the Sequence Initiator shall not transmit any Data frames within that range within a timeout period of R_A_TOV. Both the Sequenc Initiator and Sequence Recipient discard frames within that range. After R_A_TOV has expired, the Sequence Initiator shall reinstate the Recovery_Qualifier using a Reinstate Recovery Qualifier Link Service request.

29.2.4.2 Class 3

In Class 3, the expected event for the Recipient is another Data frame for the same Sequence. Other events such as Abort Exchange shall also stop the Sequence timer. When a Sequence Recipient receives the last Data frame transmitted for the Sequence, it shall verify that all frames have been received.

If the Sequence Initiator periodically transmits an ABTS frame, the Sequence Recipient is able to inform the Sequence Initiator of the last deliverable Sequence. If the Sequence Initiator does not transmit ABTS frames, in Discard multiple Sequences Exchange Error Policy following an error in a Single Sequence, the Sequence Recipient shall continue to abnormally terminate subsequent Sequences and not deliver them to the FC-4 or upper level due to the requirement of in-order delivery of Sequences in the order transmitted. For bidirectional Exchanges, it is possible to infer proper Sequence delivery without the use of ABTS, if Sequence Initiative has been transferred and the reply Sequence for the same Exchange is received.

29.2.4.3 End-to-end Class 2 Credit loss

In Class 2 it is possible to reduce end-to-end Credit to zero as a result of one or more Sequence timeouts. The Link Credit Reset (LCR) Link_Control frame shall be transmitted by the N_Port which has no end-to-end Credit (Sequence Initiator) to the destination N_Port (Sequence Recipient). The Sequence Initiator (S_ID of LCR frame) shall perform normal recovery for the Sequence that timed out (see 24.6.2).

F_BSY may be returned by the Fabric if unable to deliver the LCR frame. A Reject may also be returned if either the S_ID or D_ID is invalid or an invalid delimiter is used.

When an N_Port receives LCR, it shall discard the Data in its buffers associated with the S_ID of the LCR frame and abnormally terminate the Sequences associated with any discarded frames.

29.2.5 OLS transmit time ut

When a Port is performing the Online to Offline Protocol (see 16.6.3), the Port shall timeout the transmission of OLS (5 ms) before reception of a Primitive Sequence in response. The timeout value of 5 ms allows a Port to enter the Offline State in the absence of an appropriate response from the attached Port.

29.3 Link error detection

Link errors are detected at three levels.

29.3.1 Link Failure

The first level of link error detection is at the receiver. Link Failure is detected under the following conditions:

- Link Failure conditions (see 29.2.2), or
- Reception of the NOS Primitive Sequence (see 16.4.2).

Recovery from Link Failure is accomplished by performing the Link Failure Protocol (see 16.6.4).

29.3.2 Code violations

Code violations are link errors which result from an invalid transmission code point or disparity error. If a code violation occurs during Idles, it is recorded in the Link Error Status Block by incrementing the Invalid Transmission Word count by one. If a code violation occurs during frame reception (see 17.8), the Link Error Status Block shall also be updated by incrementing the Invalid Transmission Word count by one and the frame identified as invalid. The Data Field of the invalid frame may be discarded or processed based on the Exchange Error Policy.

NOTE - When a code violation is detected, the actual location of the error may precede the location at which the code violation is detected (see figure 40). In particular, even if the code violation is detected following the Frame_Header, fields in the header may not be valid.

29.3.3 Primitive Sequenc prot c l error

If an N_Port is in the Active State and it receives the Link Reset Response Primitive Sequence (LRR), it shall detect a Primitive Sequence protocol error which is counted in the Link Error Status Block (LESB).

29.4 Link error recovery

The Link recovery hierarchy is shown in figure 82.

The recovery protocols are nested and organized from the most serious to least serious link action.

- Link Failure Protocol (see 16.6.4)
- Link Initialization Protocol (see 16.6.2)
- Link Reset Protocol (see 16.6.5)

Primitive Sequence Protocols provide two basic functions. The first function is to notify the other end of the link that a specific type of link error has occurred. The second function is to reset the link to a known state at both ends.

Link initialization and shutdown

Link initialization is accomplished by performing the Link Initialization Protocol. When this protocol is complete, the Port is synchronized, transmitting and receiving Idles (see 16.6.2) and is in the Active State.

Shutdown prior to power-off is accomplished by the Online to Offline Protocol. This Protocol provides an attached Port with a graceful indication prior to loss of signal. This avoids logging an error event for a normal system function (see 16.6.3).

Link Reset Protocol

Link Reset Protocol may be performed when any of the following conditions are detected:

- Connection Recovery (see 28.8),
- Link timeout (29.2.3),
- Buffer-to-buffer overrun (i.e., an N_Port receives a frame subject to buffer-to-buffer flow control without a buffer available).

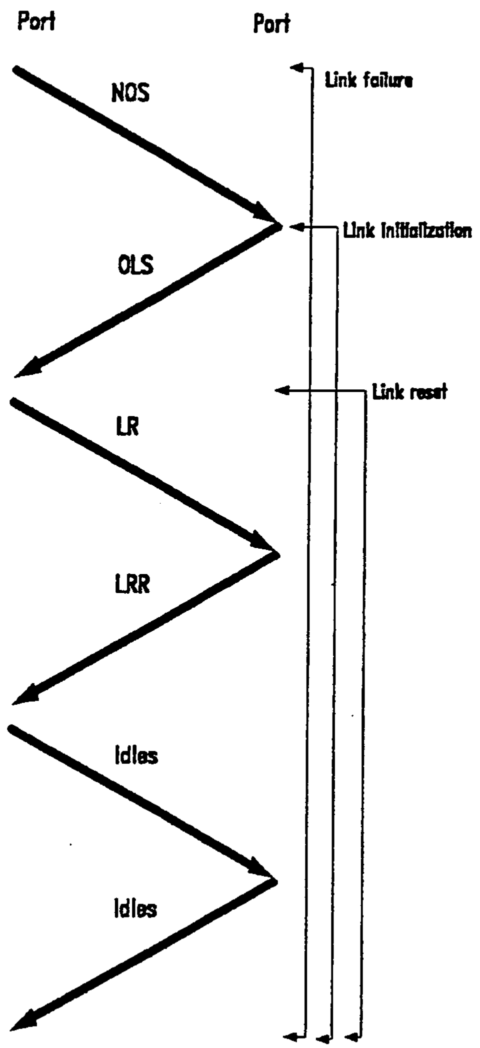


Figure 82 - Link recovery hierarchy

29.5 Link recovery - secondary effects

If the recovery action from an error involving link integrity requires transmission of a Primitive Sequence, Sequences within an Exchange may be adversely affected based on Class of Service.

Class 1
During a Class 1 Dedicated Connection, transmission or reception of Primitive Sequences removes the connection. While an N_Port is transmitting a Primitive Sequence, it shall discard any subsequent Class 1 frame received. For intermix, see the discussion regarding Class 2 and Class 3. Aft r receiving the first Primitive Sequence of any Primitive Sequenc Protocol, the end-to-end Credit and buffer-to-buffer Credit shall be reset to their original Login values in both N_Ports and the F_Ports involved.

After leaving the Active State to perform a Primitive Sequence Protocol, all frame Sequences that were Active are terminated abnormally. Recovery on a Sequence by Sequence basis is required of the Sequence Initiator at the discretion of the FC-4 or upper level. The Exchange is not aborted or abnormally terminated, unless ABTX or LOGO is explicitly performed by either the Originator or Responder. When Sequences are abnormally terminated, the current status of each Sequence is posted in the corresponding Exchange Status Block and the Sequence Status Blocks are reset and released. Link resources associated with the Active Sequences are released. If the Sequence Initiative is in doubt, the previous Initiator N_Port shall interrogate the other N_Port's Exchange Status Block to determine which N_Port has the Sequence Initiative (see 29.7.1).

Class 2
When Primitive Sequences are transmitted or received, the F_Port may discard any Class 2 frames held in its buffers. While an N_Port is transmitting a Primitive Sequence, it may discard any subsequent Class 2 frame received. After receiving the first Primitive Sequence of any Primitive Sequence Protocol, the buffer-to-buffer Credit shall be reset in the N_Port and the F_Port. Both the N_Port and F_Port may begin transmitting frames after entering the Active State.

Active Sequences within an Exchange are not necessarily affected. Therefore, normal processing continues and Sequence recovery is performed as required.

Class 3
When Primitive Sequences are transmitted or received, the F_Port may discard any Class 3 frames held in its buffers. While an N_Port is transmitting a Primitive Sequence, it may discard any subsequent Class 3 frame received. After receiving the first Primitive Sequence of any Primitive Sequence Protocol, the buffer-to-buffer Credit shall be r set in the N_Port and th F_Port. Both the N_Port and F_Port may begin transmitting frames after entering the Active State.

Active Sequences within an Exchange are not necessarily affected. Therefore, normal processing continues and Sequence recovery is performed as required.

29.6 Exchange Integrity

Exchange Integrity is accomplished by proper Exchange and Sequence management.

Applicability

Since Class 3 does not use ACK or Link_Response frames, Sequence integrity is verified at the Sequence Recipient on a Sequence by Sequence basis. In Class 3, only the Recipient is aware of a missing frame condition and communication of that information to the Initiator is the responsibility of the FC-4 or upper level.

The remaining discussion concentrates on Class 1 and 2. Items applicable to Class 3 shall be specified explicitly.

29.6.1 Exchange management

An Exchange is managed according to the rules specified in 24.3.1 When an Exchange is originated, the Originator specifies the Exchange Error Policy for the duration of the Exchange. Delivery of Data within a Sequence from the Originator to the Responder or from the Responder to the Originator shall be in the same order as transmitted. The discarding of Sequences, the delivery order of Sequences, and the recovery of Sequences varies based on the Exchange Error Policy identified by the Originator (see 18.5 Abort Sequence Condition bits (F_CTL bits 5-4).

29.6.1.1 Exchange Error Policies

There are two fundamental Exchange Error Policies - discard and process. Discard policy means that a Sequence is delivered in its entirety or it is not delivered at all. There are three variations of discard policy which relate to the deliverability of ordered Sequences and one method of retransmission at the FC-2 level. Process policy allows an incomplete Sequence to be deliverable. Process policy allows the Data portion of invalid frames to be delivered if the Sequence Recipient has reason to believe that it is part of the proper Exchange. Se 24.3.10 for rules which discuss detailed requirements for each type of Discard and Process policy.

Discard multiple Sequences
The Discard multiple Sequences Error Policy requires that for a Sequence to be deliverable, it shall be complete (all Data frames received and accounted for) and any previous Sequences, if any, for the same Exchange from the Sequence Initiator are also deliverable. This policy is useful if the ordering of Sequence delivery (Sequence A followed by Sequence D, followed by Sequence T, and so forth) is important to the FC-4 or upper level. Sequences shall be delivered to the FC-4 or upper level on a Sequence basis in the same order as transmitted.

Discard a single Sequence
The Discard a single Sequence Error Policy requires that for a Sequence to be deliverable, it shall be complete (all Data frames received and accounted for). There shall be no requirement on the deliverability of previous Sequences for the Exchange. This policy is useful if the Payload of the Sequences delivered contains sufficient FC-4 or upper level information to process the Sequence independently of other Sequences within the Exchange. Sequences shall be delivered to the FC-4 or upper level on a Sequence basis in the same order as received.

Process with infinite buffering
The Process with infinite buffering Error Policy does not require that a Sequence be complete or that any previous Sequences be deliverable. Process policy allows an N_Port to utilize the Data Field of invalid frames under certain restrictive conditions (see 17.8.1 and 17.8.2). The Process with infinite buffering Error Policy is intended for applications such as a video frame buffer in which loss of a single frame has minimal effect or no effect on the Sequence being delivered. Frames shall be delivered to the FC-4 or upper level in the same order as transmitted.

Discard multiple Sequences with retransmission
The Discard multiple Sequences with immediate retransmission is a special case of the Discard multiple Sequences Error Policy. This policy is only applicabl to an Exchange where all transmission is in Class 1. It provides a method by which the Sequence Recipient may request immediate Sequence retransmission at the FC-2

level (see 29.7.1.2) under the guidance or authorization of the FC-4 or upper level. Sequences shall be delivered to the FC-4 or upper level on a Sequence basis in the same order as transmitted.

29.6.2 Sequence integrity

Sequence management and integrity involves

— proper initiation of Sequences (see 24.3.4),

— proper control of the ordering of Sequences (SEQ_ID usage) with continuously increasing SEQ_CNT (see 24.3.6),

— proper control of Data frames by SEQ_CNT within single Sequence (see 24.3.6), and

— proper completion of a Sequence (see 24.3.8).

Frame identification and Sequence identification (see 18.1.1 and 18.1.2) provide the appropriate uniqueness to ensure the integrity of the Data delivered. A Sequence Recipient shall not reassemble and deliver the Data frames of a single Sequence unless all of the Data frames adhere to the Sequence management rules specified in 24.3.5.

29.6.3 Sequence error detection

Sequence errors are detected in three ways

a) detection of a missing frame (see 24.3.10 and 24.3.11),
b) detection of a Sequence timeout (see 29.2.4), and
c) detection of rejectable condition within a frame (see 20.3.3.3).

In Class 1 and 2, detection of Sequence errors by the Recipient is conveyed in the Abort Sequence Condition bits (F_CTL 5-4) in an ACK frame or by a P_RJT frame. Otherwise, either the Sequence Initiator or Sequence Recipient or both detect a Sequence timeout.

Exchange and Sequence status are tracked in the Exchange Status Block (see 24.8.1 and 24.3.14) and the Sequence Status Block (see 24.8.2 and 24.3.12), respectively.

29.6.4 X_ID processing

During certain periods in an Exchange, one or both X_ID fields may be unassigned. If an X_ID is unassigned, special error recovery for both the Sequence Initiator and the Sequence Recipient may be required (see 25.4).

29.7 Sequence recovery

Sequence recovery is under control of the individual FC-4 or upper level as well as options within a specific implementation. Such control may be exercised in the form of guidance, authorization to automatically perform recovery, a requirement to inform the upper level prior to recovery, or no Sequence recovery within the Exchange encountering a Sequence error. FC-PH specifies procedures to terminate or abort a Sequence, recover end-to-end Credit, handle missing or delayed frames, and synchronize both N_Ports with respect to Sequence and Exchange status. FC-PH does not require Sequence retransmission within the same Exchange in which a Sequence error is detected.

29.7.1 Abnormal Sequence termination

There are multiple methods by which a Sequence completes abnormally and one method by which a Sequence completes but is only partially received by the Sequence Recipient. When a Sequence completes abnormally, it shall not be delivered to the FC-4 or upper level. The rules for normal Sequence completion are specified in 24.3.8. The methods by which a Sequence completes abnormally include:

— the Sequence Initiator aborts the Sequence with an ABTS frame utilizing the Abort Sequence Protocol. At the time the ABTS frame is received, one or more Sequences are incomplete.

— a Class 1 Dedicated Connection is broken by transmission or reception of a Primitive Sequence while the Sequence is Open.

— if the Exchange of which a Sequence is a member is abnormally terminated, each Open Sequence shall also be abnormally completed (see 24.3.13).

— Logout (see 21.4.8).

A Sequence may complete normally with only a part of the Sequence b ing received by th

Sequence Recipient in the Stop Sequence Protocol.

29.7.1.1 Abort Sequence Protocol

The Sequence Initiator shall begin the Abort Sequence Protocol (ABTS Protocol) by transmitting the ABTS Basic Link Services frame. The SEQ_ID shall match the SEQ_ID of the last Sequence transmitted even if the last Data frame has been transmitted. The ABTS frame may be transmitted without regard to whether transfer of Sequence Initiative has already been attempted or completed. The SEQ_CNT of the ABTS frame shall be one greater than the SEQ_CNT of the last frame transmitted for this Exchange by the Sequence Initiator of the ABTS frame.

The Sequence Recipient shall accept an ABTS frame even if the Sequence Initiative has been previously transferred. The Recipient determines the last deliverable Sequence as the Recipient for this Exchange and it includes that SEQ_ID in the Basic Accept Payload along with a valid indication (see 21.2.2). The Payload of the Basic Accept also includes the current OX_ID and RX_ID for the Exchange in progress. Low and high SEQ_CNT values are also returned. The low SEQ_CNT value is equal to the SEQ_CNT of the last Data frame (End_Sequence = 1) of the last deliverable Sequence. If there was no deliverable Sequence, the low value is zero.

The high SEQ_CNT value shall match the SEQ_CNT of the ABTS frame. The combination of OX_ID, RX_ID, low SEQ_CNT and high SEQ_CNT defines a Recovery_Qualifier range. This range indicates a range of Data frames that were not and shall never be delivered to the FC-4 or upper level in the Discard multiple Sequences Error Policy. In the Discard a single Sequence Error Policy, the Recovery_Qualifier may contain Sequences which have been delivered.

If the ABTS frame is transmitted in Class 2 or 3, the Recovery_Qualifier shall be timed out by the Sequence Initiator of the ABTS frame for a timeout period of R_A_TOV. After the R_A_TOV timeout has expired, th Sequence Initiator of the ABTS frame shall issue a Reinstat Recovery Qualifier Link Service r quest in order to purge the Recovery_Qualifier. Timing out the Recovery_Qualifier range for R_A_TOV allows both N_Ports to discard frames received in this range. This ensures the Data integrity of the Exchange.

NOTE - Since streaming Sequences across multiple Classes for the same Exchange is not allowed, and since ABTS shall be transmitted as part of the last Sequence transmitted, the Abort Sequence Protocol is performed in the same Class of Service in which the error, if any, occurred. Therefore, there is no ambiguity between the Sequence Initiator and the Sequence Recipient as to whether a Recovery_Qualifier is needed. If ABTS is transmitted in Class 1, then none is needed. If ABTS is transmitted in Class 2 or Class 3, then a Recovery_Qualifier is established. For example, if a Sequence Initiator has transmitted a Sequence in Class 1, but Link Recovery is performed before all acknowledgements are received, then the Sequence Initiator is required to reestablish a Class 1 Connection in order to transmit ABTS.

Transmission of BA_ACC by the Sequence Recipient is a synchronizing, atomic event. The Sequence Recipient shall discard any frames received within the Recovery_Qualifier range, if timeout is required, the instant that the Basic Accept is transmitted and thereafter, until it receives a Reinstate Recovery Qualifier (RRQ) Extended Link Services request. The Sequence Initiative F_CTL bit setting in the BA_ACC shall indicate whether the Sequence Initiative is held or transferred to the Sequence Initiator of the ABTS frame. If Sequence Initiative is held by the Sequence Recipient of the ABTS frame, it shall withhold Sequence Initiative transfer until the ACK to the Basic Accept is received.

In like manner, after the Sequence Initiator has received the Basic Accept to the ABTS frame, it shall discard any frames received for the Recovery_Qualifier range. In Class 2 and 3, the Sequence Initiator shall retire the SEQ_CNT range within the Recovery Qualifier until R_A_TOV has expired, or it shall abort the Exchange (the Recovery_Qualifier for the Exchange times out R_A_TOV).

When the Basic Accept has been received by the Sequence Initiator, it may reclaim any outstanding end-to-end Credit for all unacknowledged Data frames within the SEQ_CNT range of the Recovery_Qualifier. After the Sequence Initiator of the ABTS frame has received the BA_ACC with Sequence Initiative transferred to the Initiator, it may retransmit the S quences that it determines as non-deliverable by the S quence Recipient (see 24.3.8 and 24.3.11).

If a Recovery_Qualifier exists and is being timed out (R_A_TOV) and another Sequence error occurs which would cause transmission of the ABTS frame, the Exchange shall be aborted using the ABTX request. However, if the Reinstate Recovery Qualifier request has been completed such that the Recovery_Qualifier has been purged, the ABTS Protocol may be utilized and a new Recovery_Qualifier may be established.

Special case - new Exchange

If a Sequence Initiator of the ABTS frame attempts to originate a new Exchange and a Sequence timeout occurs, the Sequence Initiator shall transmit the ABTS frame as in the ABTS protocol defined. If the Sequence Recipient receives an ABTS frame for an Exchange which is unknown, it shall Open an Exchange Status Block, with OX_ID = value of ABTS, RX_ID = hex 'FFFF', and post the SEQ_ID of the ABTS frame. The Basic Accept Payload shall indicate invalid SEQ_ID, a low SEQ_CNT set to zero, and a high SEQ_CNT set to SEQ_CNT of the ABTS frame.

The Sequence Initiator of the ABTS frame shall timeout the Recovery_Qualifer, if required, and transmit the Reinstate Recovery Qualifier in the normal manner.

Special case - Class 3

If a Sequence Initiator desires to determine the status of an Open Exchange in Class 3, it is possible to transmit an ABTS frame on a periodic basis in order to determine successful status (see 24.3.8). Deliverable Sequences are not affected, yet undeliverable Sequences are detected by the Sequence Initiator. However, it should be noted that if the ABTS frame arrives prior to a frame in flight for the Active Sequence, an otherwise deliverable Sequence becomes undeliverable once a Recovery_Qualifier is established. If the ABTS Protocol is performed, the normal rules are applicable.

29.7.1.2 Class 1 Sequence retransmission

In an asynchronous environment, Sequence retransmission is made difficult due to a possible difference in error information available in the Sequence Recipient compared to the Sequence Initiator. Therefore, there are restrictions placed on its use:

- Class 1 only (in-order delivery, with no frames delayed in the Fabric).
- the Sequence Initiator shall request Discard multiple Sequences with immediate retransmission as the Error Policy on the first Data frame of the Exchange.
- the Sequence Recipient shall not be required to support Sequence retransmission and may reply with Abort Sequence Condition bits set to (0 1) if a missing frame error is detected.
  - if the Sequence Recipient does not support the Discard multiple Sequences with immediate retransmission Error Policy, it shall set the Abort Sequence Condition bits (F_CTL 5-4) to (0 1) to indicate a missing frame error.
  - if the Sequence Recipient does not support the Discard multiple Sequences with immediate retransmission Error Policy, it shall transmit an ACK with the Abort Sequence Condition bits (F_CTL bits 5-4) set to (0 1) if a Data frame is received with the retransmit bit (F_CTL bit (9) set = 1.
- if Discard multiple Sequences with immediate retransmission is supported by the Sequence Recipient, the Sequence Recipient shall set the Abort Sequence Condition bits (F_CTL 5-4) to (1 1) to indicate a missing Data frame error when:
  - the Sequence Recipient determines that a missing frame error has occurred in a Sequence such that an ACK may be transmitted for the same SEQ_ID of the first non-deliverable Sequence with the retransmit bit in F_CTL set = 0.
  - the Sequence Recipient shall continue to transmit ACKs for the same SEQ_ID with the Abort Sequence Condition bits set = (1 1).
  - for any subsequent streamed Sequences, the Sequence Recipient shall set the Abort Sequence Condition bits to (0 1) and the

Sequences shall be non-deliverable. The Sequence Recipient shall discard those subsequent Sequences with no Sequence Status Blocks opened, and no information saved.

—the Sequence Recipient shall save the SEQ_ID and starting SEQ_CNT for that SEQ_ID for comparison to a retransmitted Sequence (i.e. the SEQ_CNT of the **SOFi1**. frame).

—if the Sequence Recipient detects a missing Data frame error, but is unable to determine what the SEQ_ID of the first non-deliverable Sequence should be, or it is unable to determine the starting SEQ_CNT, it shall set the Abort Sequence Condition bits (F_CTL bits 5-4) to (0 1) in the corresponding ACK frame.

—when the Sequence Initiator receives an ACK with Abort Sequence Condition bits set = (1 1), it shall make a determination as to whether the SEQ_ID of the ACK frame identifies the first non-deliverable Sequence based on its ACK frame history of prior Sequences. If the Sequence Initiator is unable to make such a determination, it shall either Abort the Sequence with ABTS, Read Exchange Status, or Read Sequence Status to determine what Sequence was in error.

—if the Sequence Initiator is able to determine that the first non-deliverable Sequence in error matches the SEQ_ID of the ACK frame with the Abort Sequence Condition bits set to (1 1), it shall begin Sequence retransmission with the following restrictions:

—F_CTL bit 9 (retransmission bit) shall be set to one on each Data frame of the Sequence being retransmitted as well as any subsequent streamed Sequences until the first retransmitted Sequence is determined by the Sequence Initiator to be deliverable.
—the SEQ_ID shall match the SEQ_ID of an ACK frame with the Abort Sequence Condition bits set to (1 1),
—the SEQ_ID being retransmitted is the first non-deliverable Sequence as determined by the Sequence Initiator,
—the SEQ_CNT shall match the SEQ_CNT of the first Data frame (**SOFi1**) of the original non-deliverable S quence transmitted as determined by the Sequence Initiator.

—the Sequence Recipient shall ch ck for an ABTS frame and perform the normal Abort Sequence Protocol if received, independently of the retransmit bit (F_CTL bit 9) setting.

—the Sequence Recipient shall also check for an **SOFi1**, with F_CTL bit 9 = 1, SEQ_ID = first non-deliverable Sequence, and SEQ_CNT set to first SEQ_CNT of the non-deliverable Sequence.

—the Sequence Recipient shall set the retransmit bit (F_CTL bit 9) = 1 in the ACK frame corresponding to a Data frame in which F_CTL bit 9 = 1.

—if the Sequence Recipient detects an error in the restart point of the Sequence retransmission (either incorrect SEQ_ID, or incorrect SEQ_CNT of **SOFi1**.), it shall transmit an ACK with the retransmission bit (F_CTL bit 9) set to 1, the Abort Sequence Condition bits set to (0 1), requesting ABTS, and the Sequence being retransmitted shall be non-deliverable.

—after the Sequence Initiator determines that the retransmitted Sequence is successfully delivered (ACK **EOFt**), it shall transmit subsequent Sequences with F_CTL bit 9 = 0.

29.7.1.3 Recipient abnormal termination

If no Data frames are being received for a Sequence in error, the Sequence Recipient shall timeout the Sequence and abnormally terminate the Sequence by setting status in the Sequence Status Block to indicate Sequence timed-out by Recipient, update the Sequence status in the Exchange Status Block, and release link facilities associated with the Sequence. If an ABTS frame for the abnormally terminated Sequence is received, the Abort Sequence Protocol shall be performed and completed.

The Sequence Recipient may timeout the Exchange in a system dependent manner and timeout period.

29.7.1.4 End_Sequence

If the last Data frame of a Sequence has been transmitted and the Sequence Initiator detects a Sequence timeout, the Initiator may initiate an Exchang with a Read Exchange Status Link Service request to determine S qu nce and Exchange status. If the Initiator is in the process of timing out a Sequence for a missing **EOFt** with Sequence Initiative transferred and it receives a new Sequence initiated by the Recipient (new

Initiator), it shall assume that the previous Sequence ended successfully. In order to make such an assumption, the N_Port Identifier, OX_ID, and RX_ID shall be the same for the new Sequence as the Sequence which transferred Sequence Initiative.

From a Recipient view, if the last Data frame is lost, the Recipient abnormally terminates the Sequence when a Sequence timeout is detected.

29.7.2 Stop Sequence Protocol

The Stop Sequence Protocol shall be used when a Sequence Recipient wishes to terminate a Sequence without invoking a drastic recovery protocol. To cause a Sequence to be terminated by the Initiator, the Sequence Recipient shall set the Abort Sequence Condition bits in F_CTL to a 1 0 value in the ACK to each Data frame received after the Recipient recognizes the need to terminate the Sequence. When the Sequence Initiator receives the first ACK with the Stop Sequence Condition indicated, it shall terminate the Sequence by transmitting a Data frame of the Sequence with End_Sequence = 1. If the Sequence Initiator has already transmitted the last Data frame of the Sequence, no further action is required of it except that which may be required by the FC-4 or upper level.

Once the Sequence Recipient has indicated the Stop Sequence condition, it shall not report Sequence errors related to Data frames with a SEQ_CNT greater than that of the Data frame at which the Stop Sequence condition was recognized. However, it shall observe the normal Sequence timeout protocols before transmitting the ACK to the frame with the End_Sequence bit set and shall recover Credit in the normal manner.

NOTES

1 When the Sequence Initiator stops the Sequence, or if it sent the last Data frame before receiving the Stop-Sequence indication, it may either hold or pass Sequence Initiative as determined by the Upper Level Protocol. It is the responsibility of the Upper Level Protocol to define the protocol for indicating to the Sequence Initiator why the Sequence was stopped, if such an indication is needed, and the protocol for transferring Sequence Initiative if needed.

2 A common use of this protocol is to signal that there is no more room in the Upper Level Protocol buffer for the Data being received. To terminate the Sequence when the Upper Level Protocol buffer is exhausted, the Sequence Recipient stores as much data as possible from the first frame whose Payload cannot be completely stored and indicates the Stop Sequence condition in the Abort Sequence Condition bits in F_CTL in the ACK to that Data frame and in each subsequent ACK until the end of the Sequence. When the Sequence ends, the ULP protocol may send a message from the Sequence Recipient to the Initiator which includes the count of the number of bytes in the Sequence which were stored before the ULP buffer was exhausted.

29.7.3 End-to-end Credit loss

FC-PH does not specify the method to be employed for Credit allocation to a destination N_Port. If destination N_Port end-to-end Credit is allocated on a Sequence basis, then that portion of end-to-end Credit is reclaimed when the Sequence is aborted or abnormally terminated. When a Sequence is aborted, any end-to-end Credit for outstanding ACK frames associated with that Sequence may be reclaimed. This applies to both Class 1 and Class 2.

29.8 Link Error Status Block

The errors shown in figure 83 are accumulated over time within an N_Port. The format shown is the format in which the LESB information shall be supplied in response to an RLS Link Service command. It does not require any specific hardware or software implementation. The errors accumulated provide a coarse measure of the integrity of the N_Port link over time. No means is provided to reset the LESB, it merely is allowed to overflow and thus reset. The counts shall be 32 bit binary integers.

An N_Port may choose to log these events as well as other errors which occur on an N_Port specific basis which is not defined within FC-PH.

NOTE - It is recommended that F_Ports also maintain an LESB and accumulate error events in a manner which is not defined in FC-PH.

	Bits		
Word 0	31	Link Failure Count	0
Word 1	31	Loss of Synchronization Count	0
Word 2	31	Loss of Signal Count	0
Word 3	31	Primitive Sequence Protocol Error	0
Word 4	31	Invalid Transmission Word	0
Word 5	31	Invalid CRC Count	0

Figure 83 - Link Error Status Block format for RLS command

NOTE - Informative guideline to manage LESB is provided in annex V.

29.9 Detailed error detection / actions

29.9.1 Errors detected

Table 112 lists 10 categories of errors which are detectable. The categories specified relate directly to link integrity or data integrity as previously discussed. This list is representative of the types of errors which may be detected. It is not an exhaustive list. Column 1 of the table specifies a general error category. Column 2 identifies specific errors within that general category. Column 3 identifies the action which the Sequence Initiator takes on ACK frame errors detected for Sequences being transmitted or link integrity errors (ACK frame reception is only applicable to Class 1 and 2). Column 4 identifies the action which the Sequence Recipient takes on Data frame errors detected for the Sequences being received or link integrity errors.

Table 112 - Detail d rrors and actions			
Error Category	Specific Error	Seq Init Action	Seq Recp Action
Link Failure	a) Loss of Signal b) Loss of Sync > timeout period	a) 12 b) 12	a) 12 b) 12
Link Errors	a) Invalid Transmission Word during frame reception b) Invalid Transmission Word outside of frame reception c) Loss of Sync	a) 1,11 b) 11 c) 11	a) 1,11 b) 11 c) 11
Link Timeout	a) Class 1: all Sequences timed out b) missing R_RDYs (buffer-to buffer credit=0)	a) 6 b) 6	a) 6 b) 6
Link Reset protocol timeout	a) missing LRR response to LR transmission b) missing Idle response to LRR transmission	a) 12 b) 12	a) 12 b) 12
Sequence Timeout	a) timeout during Sequence b) timeout at end of Sequence	a) 8,10 b) 8,14,10	a) 9 b) 9
Delimiter Errors	a) Class not supported b) Delimiter usage error (**SOF$_{c1}$** while connected) c) Abnormal frame termination d) **EOF$_a$** received e) Incorrect **SOF** or **EOF** (see tables 48, 50)	a) 2 b) 2 c) 1 d) 1 e) 1	a) 2 b) 2 c) 1 d) 1 e) 1
Address Identifier errors	a) incorrect D_ID b) incorrect S_ID	a) 2 b) 2	a) 2 b) 2
Frame_Content errors	a) CRC b) Busy frame received c) Reject frame received d) TYPE not supported e) Invalid Link Control f) Invalid R_CTL g) Invalid F_CTL h) Invalid OX_ID i) Invalid RX_ID j) Invalid SEQ_ID k) Invalid SEQ_CNT l) Invalid DF_CTL m) Exchange Error n) Protocol Error o) Incorrect length p) Unexpected Link_Continue q) Unexpected Link_Response r) Login Required s) Excessive Sequences attempted t) Unable to Establish Exchange u) RO out of bounds	a) 1 b) 5 c) 3 d) 2 e) 2 f) 2 g) 2 h) 2 i) 2 j) 2 k) 2 l) 2 m) 2 n) 2 o) 2 p) 2 q) 2 r) 2 s) 2 t) 2 u) NA	a) 1 b) 5 c) 3 d) 2 e) 2 f) 2 g) 2 h) 2 i) 2 j) 2 k) 2 l) 2 m) 2 n) 2 o) 2 p) 2 q) 2 r) 2 s) 2 t) 2 u) 2
Data Frame errors	a) buffer not available - Class 1 b) buffer not available - Class 2 c) buffer not available - Class 3 d) ABTS frame received e) missing frame error detected	a) NA b) NA c) NA d) NA e) NA	a) 2,7 b) 4 c) 1 d) 8 e) 13,7
ACK_1, ACK_N frame errors	a) Missing frame error detected b) Abort sequence indicated	a) 13,8 b) 8,10	a) NA b) NA

29.9.2 Acti ns by Initiator or Recipient

1. Discard frame

In both discard and process policy, **SOFc1**, **EOFdt**, and **EOFdti** delimiters shall be processed for both invalid and valid frames. In both discard and process policy, a frame which is terminated with an **EOFa** shall be discarded.

Discard policy

If an invalid frame is detected, the entire invalid frame shall be discarded. If a valid frame is received and a rejectable or busy condition in Class 3 is detected, the entire frame shall be discarded.

Process policy

If an N_Port is able to determine that an invalid frame is associated with an Exchange which is designated as operating under Process policy, the N_Port may process and use the Data Field at its discretion, otherwise, the entire invalid frame shall be discarded.

2. Transmit P_RJT frame

If a valid Data frame is received which contains information in the Frame_Header which is invalid or incorrect, a P_RJT frame shall be transmitted with the appropriate reason code as specified in 20.3.3.3. Reason codes are defined such that the first error detected is returned as the reason code. In Class 2, R_RDY is transmitted in response to the discarded frame.

During a Class 1 Dedicated Connection, if a Data frame is received and no buffer is available, this indicates that the transmitter has an end-to-end Credit tracking problem. The reason code in P_RJT shall indicate protocol error.

3. Process Reject

When a P_RJT or F_RJT frame is received in response to a frame transmission, the reject information shall be passed to the appropriate Upper Level Protocol in order to process. Certain errors are recoverable by taking an appropriate action, such as Login. The Sequ nce shall be aborted using the Abort Sequence Protocol, regardless of possible recovery actions. For typical non-retryable errors the Exchange shall also be aborted.

If a P_RJT or F_RJT frame is received which contains information in the Frame_Header which is invalid or incorrect, the frame shall be discarded.

4. Transmit P_BSY frame

An N_Port shall track the status of its buffers such that if a Class 2 Data frame is received and no EE_buffer is available, a P_BSY shall be returned to the transmitter of the frame. If a Class 2 Data frame is received and no BB_buffer is available, the Recipient may discard the frame without issuing a P_BSY or P_RJT. Portions of the frame other than the Frame_Header are discarded. The Frame_Header is captured in order to generate a proper P_BSY Link_Response frame.

If an N_Port receives a Class 1 connect-request frame and its internal link facilities are busy or unavailable, the N_Port responds by transmitting a P_BSY frame with the appropriate reason code (see 20.3.3.2) with **EOFdt**. Portions of the frame other than the Frame_Header are discarded. The Frame_Header is captured in order to generate a proper P_BSY Link_Response frame.

An R_RDY is transmitted in response to a Class 2 frame or a Class 1 connect-request regardless of the disposition of the received frame.

5. Process Busy

When an F_BSY or P_BSY is received in response to a Class 1 connect-request, the N_Port may:

- attempt a connect-request to a different destination N_Port if such a request is pending,
- delay a period of time before reissuing the connect-request, or
- immediately reissue the connect-request.

The Fabric shall not queue a connect-request, if the destination N_Port has responded with a P_BSY to the connect-request.

NOTE - The decision as to which action to take is dictated based on the conditions in the N_Port and the period of time lost if another busy condition is returned.

When an F_BSY or P_BSY is received in response to a Class 2 Data frame, and if the N_Port has the capability to retransmit, the N_Port shall retransmit the Class 2 Data frame within E_D_TOV of the last Data frame transmission. In order to avoid reissuing a frame for an extended number of retries an N_Port may

choose to count the number of retries and decide to shutdown communication with a specific N_Port.

When an F_BSY is received in response to an ACK frame (Class 2), the N_Port shall not retransmit the ACK frame.

6. Perform Link Reset Protocol

When an N_Port has reached a buffer-to-buffer Credit_CNT of zero and has exceeded the Link timeout period (E_D_TOV), a Link timeout is detected. In Class 1, if an N_Port has timed out all Active Sequences (E_D_TOV), a Link timeout is detected. When a Link timeout is detected, the N_Port or F_Port begins the Link Reset Protocol.

In addition, in Class 1 an N_Port may not know whether a Dedicated Connection exists or not. An N_Port may initiate the Link Reset Protocol in order to reach a known state.

7. Set Abort Sequence Bits

When a Sequence Recipient detects a Sequence error by missing frame detection or other internal processing errors, the Recipient sets the appropriate Abort Sequence in F_CTL bits 5-4.

```
0 0 = Continue Sequence
0 1 = Abort, perform ABTS
1 0 = Stop Sequence
1 1 = Immediate Sequence retransmission
      requested
```

The SEQ_CNT of the first missing frame is saved in the Sequence Status Block. Only the first error is saved in the Sequence Status Block. This information may or may not be required by the Sequence Initiator for recovery purposes.

8. Perform Abort Sequence Protocol

When a Sequence Initiator detects a Sequence error or receives an appropriate Abort Sequence Condition (0 1) in F_CTL bits 5-4 in an ACK for an Open Sequence, the Initiator shall transmit an Abort Sequence Link Service request to the Recipient and transfers Sequence Initiative in order to complete Abort Sequence processing (see 29.7.1).

When a Sequence Recipient receives an ABTS frame, it shall respond as specified in 29.7.1.1 and 21.2.2.

9. Abnormally terminate Sequence

When a Sequence Recipient detects a Sequence timeout (E_D_TOV) and no Data frames are being received for the Sequence, the Recipient shall terminate the Sequence and update the Exchange Status Block.

When a Class 1 Dedicated Connection is removed by an **EOF** delimiter or a Primitive Sequence, the N_Port shall abnormally terminate any Active Sequences (see 29.7.1).

10. Retry Sequence

When a Sequence has been abnormally terminated, the Sequence Initiator may retransmit the Sequence under guidance, authorization, or control of the FC-4 or upper level. Under guidance from the upper level, it may retransmit immediately under the Discard multiple Sequences with immediate Retransmission Exchange Error Policy in Class 1 (29.7.1.2). For other Exchange Error Policies, each individual FC-4 shall specify the procedure for Sequence retransmission, if any.

11. Update LESB

The Link Error Status Block is updated to track errors not directly related to an Exchange.

12. Perform Link Failure Protocol

The Link Failure Protocol is initiated by transmission or reception of the Not Operational Primitive Sequence.

13. Error Policy processing

When an error is detected within a Sequence, the Sequence is either processed normally (process policy), or discarded (discard policy). See 29.6.1.1 for additional information.

14. Read Status Block

Read Status Block (either Sequence or Exchange) is accomplished by originating an Exchange with the appropriate Extended Link Service request.

Annex A
(informative)
Test methods

This annex is referenced by and applies to FC-0.

This annex defines terms, measurement techniques and conditions for testing jitter and optical wave shapes. This annex deals with issues specific to Fibre Channel and is not intended to supplant standard test procedures referenced in the specifications. In cases where there are conflicts between this annex and the referenced standard test procedure this annex takes precedence.

This annex directly applies to verification of terminal equipment compliance to the Fibre Channel specification and the relevant optical interface specifications. In some instances these procedures may be applicable to measurement of a single component of the system. Component performance is outside the scope of Fibre Channel compliance but it is useful from a design viewpoint.

A.1 Active output interface

A.1.1 Optical Spectrum Measurement

The centre wavelength and spectral width (FWHM or RMS) value of the Active Output Interface can be measured as appropriate using an optical spectrum analyzer per FOTP-127. The patch cable used to couple the light from the Active Output Interface to the spectrum analyzer should be short to minimize spectral filtering by the patch cable. The output signal during the measurement should be any valid 8B/10B code pattern.

A.1.2 Optical waveform

A.1.2.1 Mask of the eye diagram (laser)

The measurement of the mask of the eye diagram is covered in 6.1.1. The measurement should be performed with traffic consisting of frames of data so that the receiving equipment may p rform its normal synchronizing operations. The frame contents should consist of an encoded pseudo random pattern whose length is at least 1 000 bytes long. The purpose of this pattern is to provide frequency components in the data which span the full frequency range of the clock recovery system. The contents of the frame should end with a positive running disparity so that the K28.5 character in the first position of the EOF delimiter will be present in the complement form from the K28.5 character in the first position of the SOF delimiter.

A.1.2.2 Pulse parameters (laser)

The rise and fall times of lasers may be measured using a sequence of K28.7 transmission characters. This constitutes a square wave at 10% of the bit rate.

The measurement system should have a low pass fourth-order Bessel-Thompson transfer function (described in 6.1.1) or equivalent. If a separate filter having a fourth order Bessel-Thompson transfer function is used, care should be taken with source and load impedances of the equipment connected to the filter. In filters constructed with common techniques the proper transfer function is obtained only when the source and load impedances are at a specified value over the frequency range of interest. Other impedance values may result in the introduction of significant waveform distortion.

A.1.2.3 Pulse Parameters (LED)

Optical waveform measurements may be measured using a DC coupled wide bandwidth opto-electronic receiver and an oscilloscope. The signal during the measurement may consist of a sequence of K28.7 data bytes. This pattern corresponds to a square wave at 10% of the Bit Rate. It is important that the receiver's frequency response, gain flatness and linearity over the range of optical power being measured be sufficient to provide accurate measurement of the 0% and 100% levels and to yield accurate optical rise and fall times and minimum distortion of the optical pulse. A minimum frequency response of five times the data rate is required. In the event of a noisy optical waveform, optical detector, or oscilloscope front end the use of a digital sampling oscilloscope

operated in the averaging mode to reduce the noise is recommended.

The Active output interface extinction ratio is a measure of the modulation depth of the optical waveform exiting the node. The extinction ratio is the ratio of voltage corresponding to the 0% level (low light) to the voltage corresponding to the 100% (high light) level.

A.1.3 Jitter measurements

The Active Output Interface jitter specifications apply in the context of a 10^{-12} bit error rate (BER). Jitter may be measured with a digital oscilloscope or a bit error rate tester (BERT) as described in annexes A.3 and A.4.

A.1.4 Power Measurement for SW laser links with the OFC System

The OFC safety system (see 6.2.3) makes direct optical power measurement with most optical power meters impossible. The simplest and recommended way to overcome this difficulty is to insert an optical power splitter into the path at the point in the link where one desires to measure the optical power and connect the power meter to the split-off branch. The total optical power in the fibre can then be calculated with a knowledge of the insertion loss associated with the splitter to power meter branch.

Alternatively, one can measure the optical power directly out of the transmitter port by using a secondary light source (i.e., LED) to mimic the handshake exchange required to activate the transmitter. This secondary light source must be connected to the receiver port of the transceiver under test, mimic the handshake algorithm, and then send a continuous idle signal for the duration of the measurement. Turning off the secondary light source deactivates the transmitter under test.

It is important to realize that both of these methods can result in laser emissions that exceed Class 1 limits established by one or more laser safety standards. Hence, only individuals who have received laser safety training should be allowed to perform such measurements, and they must avoid viewing the optical output, especially with optical aids. ANSI Z136.2 should be consulted for additional information about laser safety during manufacturing and servicing of optical fibre communication systems.

A.2 Active input interface

The source of the Active Input Interface test signal may be any optical source conforming to the worst case limits of the Active Input Interface specifications of the media under test.

The test should be performed with traffic consisting of frames of data so that the receiving equipment may perform its normal synchronizing operations. The frame contents should consist of an encoded pseudo random pattern whose length is at least 1 000 bytes long. The purpose of this pattern is to provide frequency components in the data which span the full frequency range of the clock recovery system. The contents of the frame should end with a positive running disparity so the K28.5 character in the first position of the EOF delimiter will be present in the complement form from the K28.5 character in the first position of the SOF delimiter.

A compliant node should receive the test signal over the range of conditions specified with a frame error rate that corresponds to a BER less than or equal to 10^{-12}. The requirements in clause 6 were written in terms of BER to facilitate the specification of components to be used in a particular implementation.

The characteristics of the test signal may be measured with the methods outlined in annexes A.1, A.3, and A.4. Components used in a particular implementation may also be measured with these methods.

A.3 Distortion and jitter contributions

Optical waveform distortion and jitter are measured as the deviation from the ideal time position of the signal at the 50% point of the signal. The 50% is identified as the zero crossing of the AC coupled signal. The zero level is established in absence of the signal.

Ther are two types of jitter specified in th FC specifications. The contributing factors are deterministic and random jitter. The test methods are given in annex A.4.

A.4 Dist rti n and jitter m asurem nt

In this standard there are two methods used to specify jitter. The LED based systems specify DJ and RJ separately. The laser based systems specify the eye opening at BER<10^{-12} and the DJ.

A.4.1 Measurement system

The opto-electronic measurement system described in annex A.1.2 may be used for jitter measurement. If a transmitter has an eye opening or random jitter which is sensitive to the level of optical return loss, measurements should be made with high cable plant return losses (low reflections) to simulate a long link. It is under these low receiver input signals the jitter phenomena is most important.

A.4.2 Active output interface eye opening measurement

The Active Output Interface (AOI) optical eye opening measurement involves measuring the open eye of the AOI on a bit by bit basis using a BERT (BIT Error Rate Test) test set. The BER is measured at various T_d's (decision points) within the eye pattern to insure conformance to the eye opening specification in tables 6 and 9 of the standard.

The eye opening is given by:

$$EW = |T_d(max.) - T_o| + |T_o - T_d(min.)| \qquad (A.1)$$

Where:

T_o = centre of the eye pattern,
T_d = BER decision point as referenced from T_o,
T_d (max.)= rightmost decision point, and
T_d (min.)= leftmost decision point.

For each position of T_d from T_d(min.) to T_d(max.) a BER measurement is taken, giving the probability of error at the T_d position. In effect T_d is swept across the eye pattern, measuring the probability of error at each point in the eye. Th range of T_d values that result in a BER $\leq 10^{-12}$ establishes the eye opening and the smallest range from T_o must be $\geq$ half the appropriate eye opening specification.

In practic , a BER Test set is used to generate and sweep the decision point (using the BERT clock in conjunction with a precise delay generator), to make the bit by bit error count, and to calculate the measured BER. The clock used to drive the BERT should be derived from the data stream. Clocks derived from a byte rate clock associated with a synthesizer in the transmitter tend to be synchronized with some forms of deterministic jitter and will possibly underestimate the actual jitter on the signal.

The centre of the eye (T_o) pattern is the midpoint between positioning T_d to the left and right edges of the eye to achieve a BER $> 10^{-2}$. The measured BER at T_o, T_d (max.), T_d (min.) must be $\leq 10^{-12}$ and the values of both (T_d (max.) - T_o) and (T_o - T_d (min.)) must be greater than or equal to half the appropriate eye-window specification in tables 6 and 9 of the standard. All measurements are made with a measurement system having a low pass fourth-order Bessel-Thompson transfer function (described in 6.1.1) or equivalent. If a separate filter having a fourth order Bessel-Thompson transfer function is used, care should be taken with source and load impedances of the equipment connected to the filter. In filters constructed with common techniques the proper transfer function is obtained only when the source and load impedances are at a specified value over the frequency range of interest. Other impedance values may result in the introduction of significant waveform distortion.

It is important that the BERT retiming data latch be significantly faster than the timing resolution of interest.

A common practice used to save time is to measure the eye opening at higher probabilities (e.g. 10^{-6}) and then extrapolate to the eye opening at a 10^{-12} probability.

A.4.3 DJ measurement

A digital sampling oscilloscope may be used to measure DJ with a test pattern consisting of repeating K28.5 data bytes. Synchronize the oscilloscope to the pattern repetition frequency and average the waveforms to remove the effects of random jitter and noise in the measurement system. In the 20 bit pattern there will be five positive transitions and 5 negative transitions. Measure the time of the 50 % crossing of

all 10 of the transitions. The time of each crossing is then compared to th mean expected time of the crossing and a set of ten timing variations are determined. The DJ is the range of the timing variations.

A.4.4 RJ measurements

If a digital oscilloscope is available which has the capability of measuring time statistics the RJ may be determined by applying a sequence of K28.7 data bytes. The oscilloscope is used to determine the standard deviation of the 50% crossings. The Random Jitter at 10^{-12} will be the +/- 7 sigma limit points of the distribution.

In this type of measurement the signal used to trigger the oscilloscope is very important. The trigger should occur at a rate which is the data rate divided by the synthesizer ratio of the transmitter. This synchronization rate will cause the oscilloscope to ignore any phase lock loop wander which occurs between correction signals in the synthesizer.

An alternate method is to use the information from the eye width test of annex A.4.2. The random jitter may be determined from the rate of change of BER with sampling time in the low BER region of the eye. This may be done by solving the equation:

$$\begin{aligned} BER &= \frac{1}{\sqrt{2\pi}} \int_Q^{\infty} \exp\left(-\frac{x^2}{2}\right) dx \\ &\simeq \frac{1}{Q\sqrt{2\pi}} \exp\left(-\frac{Q^2}{2}\right), \quad \text{BER} < 10^{-3} \end{aligned} \tag{A.2}$$

Q is the number of RMS jitter magnitudes for the given BER. The random jitter may then be expressed as:

$$RJ_{RMS} = \left| \frac{t_1 - t_2}{Q_1 - Q_2} \right| \tag{A.3}$$

where Q_i is evaluated at t_i and where evaluation times t_1 and t_2 are located as shown in figure A.1

In this case the clock used should be the same as used to determine the eye opening.

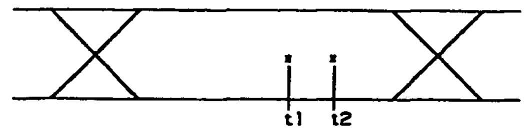


Figure A.1 - Eye diagram

A.5 Relative intensity noise (RIN) measuring procedure

This procedure describes a component test which may not be appropriate for a system level test depending on the implementation.

A.5.1 Test objective

When lasers which are subject to reflection induced noise effects are operated in a cable plant with a low optical return loss the lasers will produce an amount of noise which is a function of the magnitude and polarization state of the reflected light.

The magnitude of the reflected light tends to be relatively constant. However, the polarization state varies significantly as a function of many cable parameters, particularly cable placement. In a cable plant which is physically fixed in place the variation is slow. If the fibre is subject to motion, such as occurs in a jumper cable, the change may be sudden and extreme. The effect is unpredictable changes in the noise from the laser with the result that the communication link may exhibit sudden and unexplainable bursts of errors.

The solution to this is to assure that the lasers used do not generate excessive noises under conditions of the worst case combination of polarization and magnitude of reflected optical signal.

The noise generated is a function of the return loss of the cable plant. For the Fibre Chann l the

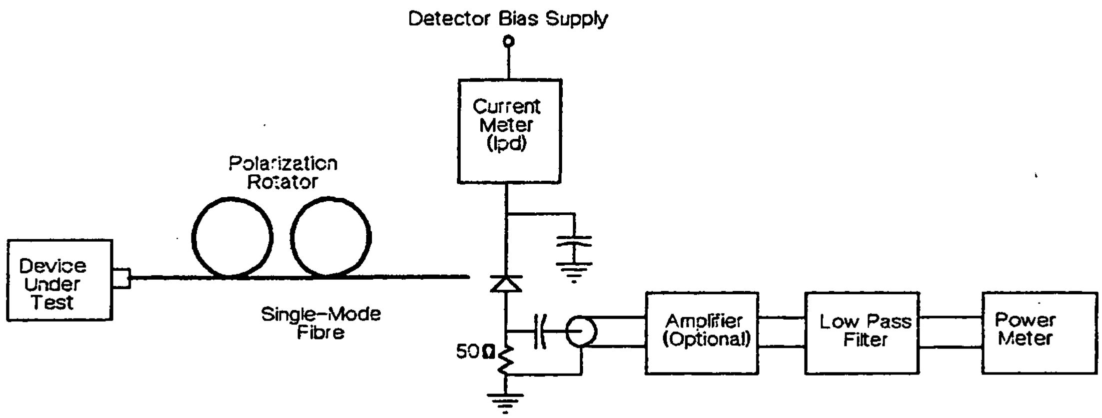


Figure A.2 - RIN test setup

specified return loss is 12 dB resulting in the notation of RIN_{12} for the relative intensity noise.

A.5.2 General test description

The test arrangement is shown in figure A.2 The test cable between the Device Under Test (DUT) and the detector forms an optical path having a single discrete reflection at the detector with the specified optical return loss. There must be only one reflection in the system as the polarization rotator can only adjust the polarization state of one reflection at a time.

Two measurements are made by the photodetector. The average optical power is determined by measuring the average current, Ipd, through the detector.

The second measurement is the noise, which is measured by AC coupling the detector into the high frequency electrical power meter. If needed, an amplifier may be used to boost the signal to the power meter.

A low pass filter is used between the photodetector and the power meter to limit the noise measured to the passband appropriate to the data rate of interest.

In order to measure the noise the modulation to the DUT must be turned off. If the laser is modulated the modulation power will be measured by the power meter rather than the noise power.

A.5.3 Component descriptions

Test Cable The test cable and detector combination must be configured for a single dominate reflection with an optical return loss of 12dB. (The Optical return loss may be determined by the method of FOTP-107) If multiple lengths of cable are required to complete the test setup they should be joined with splices or connectors having return losses in excess of 30 dB. The length of the test cable is not critical but should be in excess of 2 m.

Polarization Rotator The polarization rotator must be capable of transforming an arbitrary orientation elliptically polarized wave into a fixed orientation linearly polarized wave. A polarization rotator consisting of two quarter wave retarders will have the necessary flexibility. A possible construction technique is given in LeFevre, 1980 [3].

Detector and Detector/Amplifier The detector may be of any type which is sensitive to the wavelength range of interest. The frequency response of the detector must be higher than the cut-off frequency of the low pass filter.

The detector should be capable of determining the average and variation of the optical signal. Typically

the average optical power is determined by the current, Id, through the diode. The detector assembly should have an effective output impedance of 50 Ω.

If necessary, the noise may be amplified to a level consistent with accurate measurement by the power meter.

Filter The low pass filter should have a 3dB bandwidth of approximately 75% of the bit rate. Recommended values are shown in table A.1. The total filter bandwidth used in the RIN calculation must take the low frequency cut-off of the d.c. blocking capacitor into consideration.

Table A.1 - Filter 3 dB point

Bit Rate	Filter 3 dB Point
265,625 Mbaud	200 MHz
531,25 Mbaud	400 MHz
1,062 5 Gbaud	800 MHz

The filter should be placed in the circuit as the last component before the power meter so that any high frequency noise components generated by the detector/amplifier are eliminated. If the power meter used has a very wide bandwidth care should be taken in the filter selection to ensure that the filter does not lose its rejection at extremely high frequencies.

Power Meter The power meter should be an RF type designed to be used in a 50 Ω coaxial system. The meter must be capable of being zeroed in the absence of input optical power to remove any residual noise from the detector or its attendant amplifier, if used.

A.5.4 Test Procedure

a) Connect and turn on the test equipment. Allow the equipment to stabilize for the manufacturers recommended warm up time.

b) With the DUT disconnected zero the power meter to remove the contribution of any noise power from the detector and amplifier, if used.

c) Connect the DUT, turn on the laser and verify that the optical output power is at the average power appropriate for the DUT by means of the detector current. The laser should not be modulated.

d) Operate the polarization rotator while observing the power meter output to maximize the noise read by the power meter

e) Calculate RIN from the observed detector current and electrical noise by use of the equation:

$$RIN = 10 \log \frac{P_e}{BW\, 25\, I_{pd}^2} - G\ (dB/Hz) \qquad \text{(A.4)}$$

Where:

RIN = Relative Intensity Noise
I_{pd} = Current through the detector (A)
P_e = Electrical noise power in Watts
BW = Noise Bandwidth of the measuring system (Hz
= Low Pass Bandwidth of filter - High Pass Bandwidth of DC blocking capacitor.
25 = Effective Load Impedance (50 ohm load in parallel with a 50 ohm power meter)
G = Gain in dB of any amplifier in the Noise Measurement path

Annex B
(informative)
Diagnostic functions

A set of diagnostic functions may be provided as an implementation option for component testing of the transmitter function of the FC-0 portion of the Fibre Channel. These diagnostic functions are intended for component testing and not for test to be performed at the customer site on a shippable or production adapter.

FC-0 diagnostic functions

In order to test the FC-0 transmitter for random jitter, a compliant transmitter has to be capable of transmitting the following two test patterns:

a) A continuous sequence of D21.5 data bytes. This constitutes an alternating sequence of ones and zeros.

b) A continuous sequence of K28.7 special characters. This constitutes an alternating sequence of five ones and five zeros.

These patterns may be implemented at a bit or character level. They are to be used for transmitter testing only. The receiver may not have the capability to accept these diagnostic sequences.

Annex C
(informative)
Alternative cable plant usage

In some cases, it will be desirable to use an alternative multimode cable plant to those described in clauses 8.2 and 8.3. This may be due to the need for extended distances or for operation in locations where alternative cable plants are presently installed. These fibre types have not been studied and details for their use are not provided for in the main body of this document. Therefore, using these fibre types may change the maximum achievable distance between nodes. See annex E for methods of computing distances.

Table C.1 - Alternative fibre types

Nomenclature	Primary Subclause	Fibre Type	Distance
100-M6-SL-I	6.2	62,5μm	2m-175m
50-M6-SL-I	6.2	62,5μm	2m-350m
25-M6-SL-I	6.2	62,5μm	2m-700m
25-M5-LE-I	6.3	50μm	NOTE
12-M5-LE-L	6.3	50μm	NOTE

Note: These cases will have to be treated on an individual basis.

Annex D
(informative)
Electrical interface example

This annex describes an example implementation of the electrical interfaces for optoelectronics modules to meet the requirements of the Fibre Channel. It is recognized that individual implementers may desire to provide features in excess of this set to accomplish additional system functions or as an aid to testing.

A block diagram of the modules is shown in figure D.1. It is the option of individual implementers to place all or part of the function in a product. Because of this individual products may not have all of the interfaces exposed. Any unexposed interface is under no obligation to conform to this example.

This is an example only. Conformance to this example is not required for Fibre Channel conformance. Fibre Channel conformance is obtained by conformance to the requirements of the main body of this standard.

It is realized that differing communications media will require differing controls. For this reason this example limits itself to consideration of only the data interface and those control functions which will be common over all media. Therefore, the block labeled "Media Control" in figure D.1 will not be described in this example along with it's related signals Media_Control and Media_Status.

The example includes the majority of the function of the FC-0 layer and a portion of the FC-1 layer.

D.1 Communications Levels

Two communications levels are employed. The high speed, full data or clock rate, signals are implemented in differential ECL. The lower speed lines associated with the parallel data transfers and function controls are implemented in TTL compatible voltage levels.

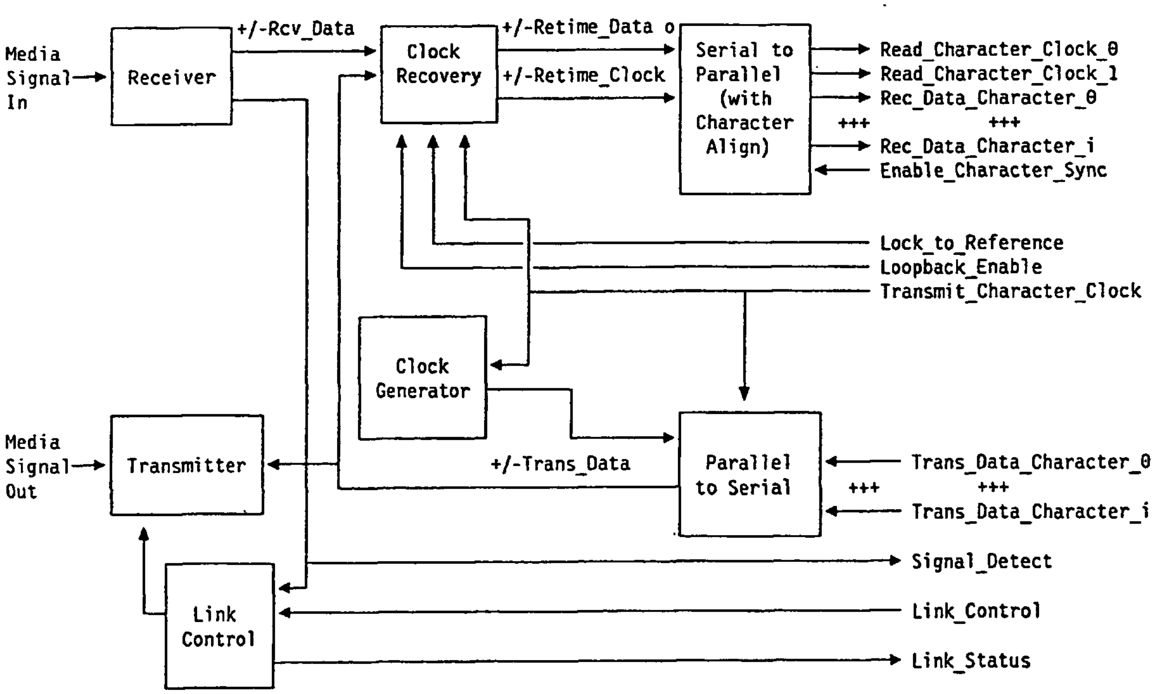


Figure D.1 - Fibre channel implementation block diagram

D.1.1 ECL

The high speed communication of the serial data and clock are implemented in differential ECL. The signal swings are shown in figure D.2 referenced to the most positive voltage driving the ECL interface circuits. This voltage could be either ground, in the case of conventional negative ECL, +5,0 V, in the case of positive referenced ECL. These outputs should be capable of driving 50 Ω transmission lines terminated in 50 Ω to 2,0 V below the reference level. The drivers and receivers are intended to operate only within the package and are not required to have the electrostatic discharge protection required for driving box to box cables.

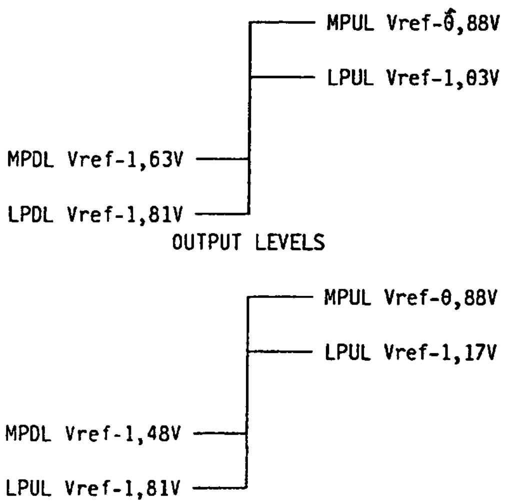


Figure D.2 - ECL communications levels

D.1.2 TTL

The parallel communications and control lines are implemented in TTL compatible voltage swings as shown in figure D.3 in order to provide compatibility with the widest number of logic technologies in the host system. Due to the relatively high speeds of the signals, especially the character clocks in the 1,063 Gbaud systems, the use of series resistors in the driver outputs is strongly recommended to minimize ringing. Because of this, the current capability of the drivers has been reduced to allow adequate output voltages in th presence of an output resistor.

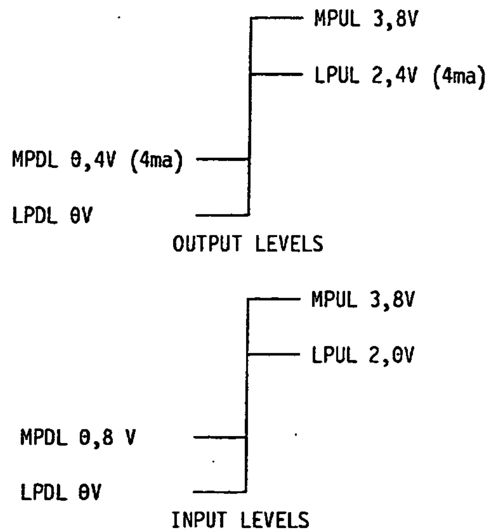


Figure D.3 - TTL compatible communications levels

D.2 Serial Interfaces

All serial interfaces use differential ECL communication levels.

D.2.1 Transmit Serial Interface

The transmit serial interface communicates the data from the serializer to the communications media. It corresponds to the FC-0_Data.Request primitive of the FC-0 Service Interface.

D.2.2 Receive Serial Interface

The receive serial interface communicates the received data from the media interface circuits to the clock extraction and retiming circuits.

D.2.3 Receive Retimed Serial Interface

The receive retimed data carries the data and recovered clock from the clock recovery circuits to the deserializer.

D.2.3.1 Retimed_Data

The Retimed_Data lines are an implementation of the FC-0_Data.Indication primitive of the FC-0 service interface.

D.2.3.2 Retimed_Clock

The Retimed_Clock lines carry the clock signal recovered from the Receive Serial Interface to the deserializer. This line is the implementation of the FC-0_Clock_Out.Indication primitive of the FC-0 service interface.

D.2.3.3 Retimed Serial Interface Timing

The relative timing between the Retimed_Data and the Retimed_Clock is shown in figure D.4.

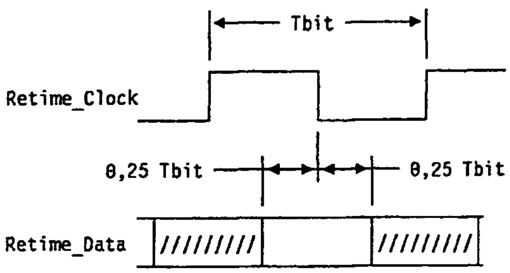


Figure D.4 - Retimed interface signal timing

D.3 Parallel Interfaces

The parallel interfaces are implemented using TTL compatible voltage levels.

D.3.1 Bus width

The example allows for bus widths of either one or two characters depending on data rate. The 133, 266 and 531 Mbaud rates utilize a one character wide bus, while the 1,063 Gbaud rate utilizes a two character bus.

D.3.2 Transmit Parallel Interface

The transmit parallel interface timings are shown in figure D.5. The data transfers are timed by the host system supplied Transmit_Character_Clock which also acts as a frequency reference for the transmit serial data and the receiver clock extraction circuitry. Data transfers are timed from the rising edge of the clock. A one character interface would use the Trans_Data_Character_0 timings and a two character interface would use both timings.

In the case of the two character interface the two data fields are to be clocked into the optoelectronic module alternately. The module timing is arranged to allow the system to switch both characters together if desired.

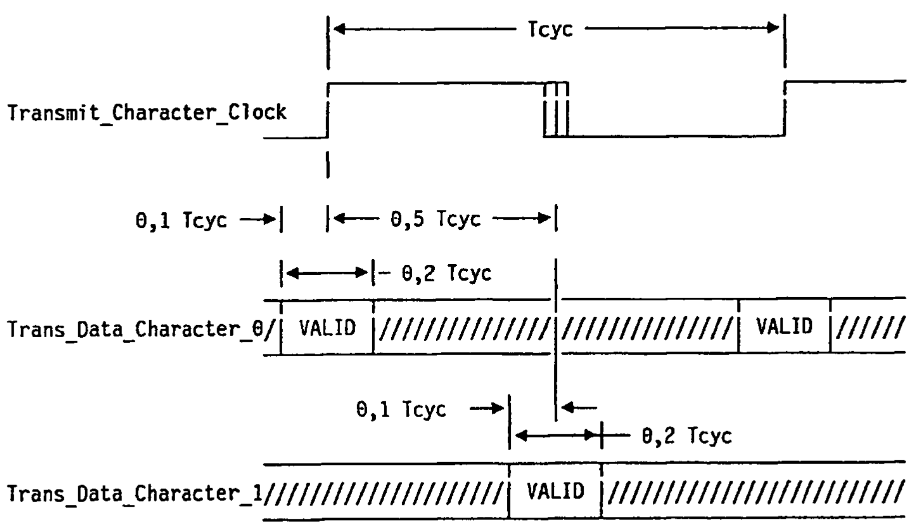


Figure D.5 - Parallel transmit data interface timing

D.3.3 Receive Parallel Int rface

The receive parallel interface timings are shown in figure D.6. In the case of a two character interface two receive character clocks are generated by the optoelectronic module to simplify system timings. The characters are presented alternately to reduce simultaneous switching noise.

In the case of a one character interface only Read_Character_Clock_0 and Rec_Data_Character_0 are required.

The data characters are valid about the negative edge of their respective Read_Character_Clocks.

D.4 Transmit Clock Reference

The Transmit_Character_Clock forms the reference for the data rate synthesizer. The timing is performed from the leading edge which must have low jitter. This line performs the function of the FC-0_Clock_Reference.Request primitive of the FC-0 Service Interface.

D.5 Control and Status Interface

The control and status lines are implemented in TTL compatible communications levels.

D.5.1 Receive Bit Sync Acquire

The receive interface bit synchronization is under control of the host system. When the FC-1 layer wishes to acquire bit synchronization the following sequence is followed:

a) Bring Lock_to_Reference line positive.

b) Wait 1,0 millisecond for the PLL to lock to the reference frequency provided by the Transmit_Character_Clock.

c) Bring Lock_to_Reference line negative.

d) Wait 2 500 bit times for the PLL to lock to the incoming data.

Should synchronization be lost while the PLL is operating near to the data frequency the clock recovery will take less than 2 500 bits to reacquire synchronization. This may occur as a result of a phase jump in the incoming data or activation of the loopback function.

This bit synchronization activity results in establishing the frequency and timing of the Retime_Clock which in turn drives the deserializer and hence the Read_Character_Clocks. In the event that the incoming signal is lost the activity of the Retime_Clock is undefined. Under these conditions the frequency or other activity of the Read_Character_Clocks is undefined.

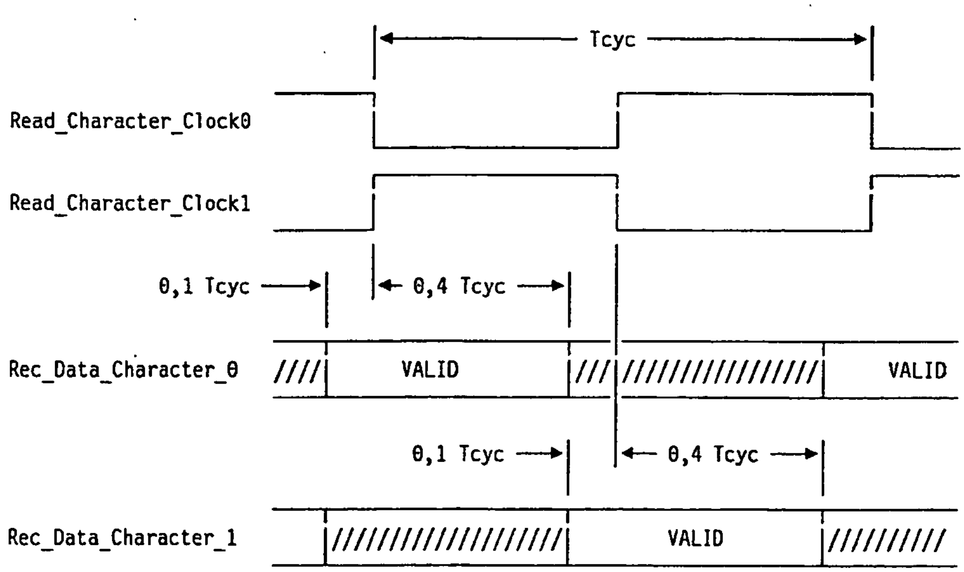


Figure D.6 - Parallel receive data interfac timing

The Lock_to_Reference line is the implementation of FC-0_Resync.Request primitive of the FC-0 service interface.

D.5.2 Receive Character Alignment

The optoelectronic module contains circuitry to provide character alignment. The circuitry will cause the output to be aligned on any occurrence of an alignment pattern in the incoming bit stream. The alignment pattern may be either a K28.5 character or just a comma. The comma is the first seven bit subset of a character composed of the patterns 0011111XXX and 1100000XXX. For additional information on character alignment requirements, see 12.1.2.2.

When alignment occurs the character of the incoming data which triggered the alignment, and all subsequent data, will be presented in uncorrupted form on the parallel data out lines. In the case of a two character interface the character which triggered the alignment data will be presented as Rec_Data_Character_0.

When the character alignment pattern is detected in the incoming data stream the Read_Character_Clocks are forced to an initial state. The state is undefined except for the following: When the clock sequence resumes there shall be at least one full half width of the Read_Character_Clock prior to the clock edge about which the data is valid. This is illustrated in figure D.7.

The occurrence of a character alignment is signaled by a pulse on the Character_Sync line. The pulse will have a nominal leading edge position coincident with the nominal rising edge timing of Read_Character_Clock0. The falling edge of the pulse will have a nominal timing coincident with the following rising edge location of Read_Character_Clock0. Due to implementation dependant timing skews the actual relative timing between Read_Character_Clock0 and

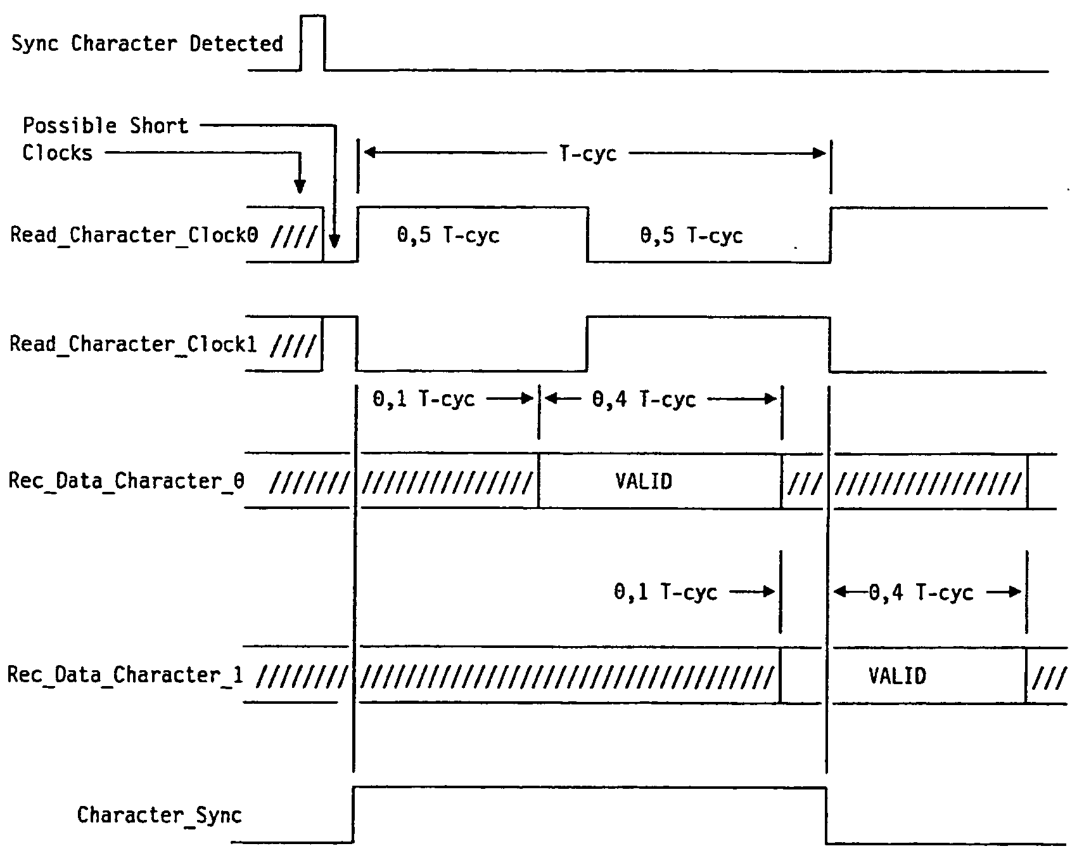


Figure D.7 - R ceiver bit sync acquire timing

Character_Sync will vary with the implementation.

This line signals the using system that the presented character is the first character of an ordered set as defined by FC-1 and that the previous clocks may have been shortened.

The alignment is enabled by holding the Enable_Character_Sync input high. This allows the character alignment to be disabled for test purposes. Alternately, the character alignment may be enabled in single acquire mode by lowering the Enable_Character_Sync input in response to a Character_Sync output.

It should be noted that if the character sync is enabled the internal circuits will respond to any occurrence of the alignment pattern in the input data stream. This may result in two anomalous situations. A single bit error may cause the alignment pattern to be falsely detected out of position. If the character sync is enabled an alignment to a false boundary will occur with the resulting destruction of all data until a properly located alignment pattern is detected. Secondly, under an open link condition the thermal noise of the receiver may cause many false locks to occur. The Character_Sync line will pulse randomly and the Read_Character_Clocks will be reset randomly.

D.5.3 Loopback Function

When the Loopback_Enable Line is brought positive the loopback function is enabled. In this operating mode the serial data stream from the transmitter is routed to the input of the retiming latch. After the time required for bit and character synchronizations this data will be available on the receive data output lines. During loopback the operation of the communications media is not defined.

The loopback function is defined at the FC-1 rather than the FC-0 level.

D.5.4 Signal Detect

The Signal_Detect line shall go positive when an incoming signal is detected. This line is the implementation of the FC-0_Signal_Detect .Indication primitive of the FC-0 service interface.

Annex E
(informative)
Cable plant usage

The data links described in clause 6 provide maximum optical power budgets. For planning purposes these data links will support the communication distances indicated in tables 6 and 9. Cable plant specification in clause 8 defines the cable plant optical attenuation ranges for the data links.

This annex documents the assumptions leading to the nominal communications distances and describes the cable plant in terms of number of connectors and splices and assumed cable parameters. Operation within this combination of parameters will provide for operation at the nominal communications distance. Other combinations will support greater or lesser distances.

E.1 Analysis method, loss limited

Using a statistical approach, the system loss budget constraint requires that the loss budget exceed the mean loss of the cable plant by at least three standard deviations.[1)]

Therefore:

$$LB > \mu_L + 3\sigma_L \qquad (D.1)$$

where
LB = Loss Budget
μ_L = Mean link loss
σ_L = Standard deviation of link loss

The link losses are given by

$$\mu_L = l_t\mu_c + N_s\mu_s + N_{CON}\mu_{CON} + \mu_{cl} \qquad (D.2)$$

and

$$\sigma_L^2 = l_R l_t \sigma_c^2 + N_s\sigma_s^2 + N_{CON}\sigma_{CON}^2 + \sigma_{cl}^2 \qquad (D.3)$$

where
l_t = Total length of the cable plant
l_R = Average reel length of the cable
μ_c = Mean cable loss
σ_c = Standard deviation of cable loss
N_s = Number of splices
μ_s = Mean splice loss
σ_s = Standard deviation of splice loss
N_{CON} = Number of connectors
μ_{CON} = Mean connector loss
σ_{CON} = Standard deviation of connector loss
μ_{cl} = Mean excess coupling loss due to mode field mismatch in the cable plant
σ_{cl} = Standard deviation of excess coupling loss

A cable splice is assumed to be located at each end of the trunk and at one per km over the length of the trunk. In addition two splices are assumed for future repair and change activity. This results in an effective reel length of 1 km. The number of splices in the link is given by

$$N_s = 3 + \frac{l_t}{1}$$

where N_s is an integer rounded upward.

E.2 Analysis method, bandwidth limited

For multimode fibre length estimates, the total distance limitation can be loss limited (see E.1 above) or bandwidth limited. There are two cases for the bandwidth limitation: short wavelength laser datalinks and LED datalinks.

For the short wavelength laser datalink, divide the cable manufacturer's specified bandwidth, given in MHz•km, by a constant factor of 0,8 to obtain the cable bandwidth in units of MBits/s•km. This 0,8 factor is a net result of combining the commonly accepted cable bandwidth to system baud rate conversion factor of 0,7 with the additional effect of chromatic dispersion present at the 780 nm wavelength

[1)] This is more stringent than the two sigma statistical treatment found in TI.106, "American National Standard for Telecommunications Digital Hierarchy Optical Interface Specification: Single-Mode," March 7,1988. This additional margin is in anticipation of a more diverse, multivendor, cable which is more difficult to control over it's life.

region. Then divide this number by the data rate, given in MBits/s to d t rmine the maximum cable plant distance.

For the LED data link, an effective cable bandwidth is calculated from the modal bandwidth and chromatic dispersion (intramodal) contributions. The length limit is then calculated from this effective cable bandwidth.

The Effective System Bandwidth can be represented by the relationship of intramodal and modal bandwidth of a given span as shown below:

$$\frac{1}{BW_{effect}^{2}} = \frac{1}{BW_{modal}^{2}} + \frac{1}{BW_{intra}^{2}} \qquad \text{(D.4)}$$

where

BW_{modal} = The bandwidth of a fibre limited by the pulse broadening due to optical power traveling via different waveguide modes. This is the specified bandwidth of the cable.

BW_{intra} = The bandwidth of a fibre limited by the pulse broadening due to chromatic dispersion (the speed of an optical pulse changes as its wavelength changes).

The chromatic dispersion of a fibre can be calculated by using the equation below:

$$D(\lambda) = \frac{S_0}{4}\left[\lambda - \frac{\lambda_0^{4}}{\lambda^{3}}\right] \text{ [ps/(nm·km)]} \qquad \text{(D.5)}$$

where

$D(\lambda)$ = Chromatic dispersion in ps/nm·km
S_0 = Dispersion function slope at λ_0
λ_0 = The zero dispersion wavelength
λ = The wavelength of operation

For LEDs operating near 1 300 nm:

$$BW_{intra} = \frac{187000}{\sigma_{intra}} \text{ [MHz]} \qquad \text{(D.6)}$$

$$\sigma_{intra} = L\sqrt{D(\lambda)^2(\Delta\lambda \times k)^2 + \frac{S_0^{2}(\Delta\lambda \times k)^4}{2}} \qquad \text{(D.7)}$$

where

σ_{intra} = RMS pulse broadening in ps
k = 0,425 (the conversion factor from FWHM to RMS)
$\Delta\lambda$ = FWHM source spectral width in nm
L = Length in km
S_0 = Dispersion slope in ps/(nm²·km)

E.3 Single-mode cable plant usage

For the single-mode cable length estimates consider the data links of 6.1, "SM data links." The data links with a nominal distance capability of 10 km have 14 dB loss budgets and the data links with a nominal distance of 2 km have 6 dB loss budgets. The connector at each end is accounted for in the interface power specifications of 6.1 and is not included in the cable plant. Two (2) dB has been removed from the difference between the transmitter output power and the receiver sensitivity specifications to account for optical path power penalties.

The cable parameters for the single-mode loss budget are specified in table E.1.

Table E.1 - Examples of Single-mode cable plant parameters.
Used for calculating loss budgets (see E.2)

FC-0 Parameter	Sym	Value
Cable	μ_c σ_c	0,50 dB/km 0 dB/km
Splice	μ_s σ_s	0,15 dB/spl 0,10 dB/spl
Connector	μ_{con} σ_{con}	0,60 dB/con 0,20 dB/con
Additional coupling loss	μ_{cl} σ_{cl}	0,4 dB 0 dB

Substituting in equations (D.2) and (D.3) gives:

$$\mu_L = 0{,}50 l_t + 0{,}15(3 + l_t) + 0{,}6N_{CON} + 0{,}40$$
$$= 0{,}65 l_t + 0{,}85 + 0{,}6N_{CON}$$

and

$$\sigma_L^2 = l_R l_t(0)^2 + (3 + l_t)(0{,}10)^2 + N_{CON}(0{,}20)^2 + (0)^2$$
$$= 0{,}01 l_t + 0{,}04N_{CON} + 0{,}03$$

The maximum lengths estimated by iteratively solving equation (D.1) are found in table E.2.

Table E.2 - Calculat d max. single-mode length			
FC-0	LB	N_{con}	Length
100-SM-LL-L 50-SM-LL-L 25-SM-LL-L	16 - 2 = 14	8	9,8 km
100-SM-LL-I 25-SM-LL-I	8 - 2 = 6	4	2,1 km

E.4 Multimode cable plant usage

The optical data links in 6.3 and 6.2 provide optical power budgets of 6 to 12 dB for SW laser links and 6 dB for LW LED links. The connector at each end is accounted for in the interface power specifications and is not included in the cable plant. In SW laser and LED based systems, there are no additional losses to account for reflection, dispersion, jitter or mode selective power penalties. However, it is common practice for multi-mode fibre data links, to not specify a separate optical path power penalty but rather to include such a penalty into the actual link budget specifications of either the transmitter or receiver. Typically for links with short coherence (self-pulsating) lasers, this optical path power penalty is small and implicitly contained in the receiver sensitivity. Alternately, lasers with longer coherence may be used. However, any optical path power penalty due to modal noise associated with using these lasers on the specified cable plant should be compensated (e.g., increasing launch power).

For planning purposes the data link will support distances shown in table E.4 based on the assumption noted in table 9. Other combinations will support greater or lesser distances.

The cable parameters for the multimode loss budget are specified in table E.3.

Table E.3 - Multimode cable plant parameters		
FC-0 Parameter	Sym	Value
Cable Versions 1) 50/125, λ=780nm 2) 62,5/125, λ=780nm 3) 50/125 and 62,5/125 λ=1 300nm	μ_{c1} μ_{c2} μ_{c3} σ_c	4,0 dB/km 4,5 dB/km 1,0 dB/km 0 dB/km
Splice	μ_s σ_s	0,15 dB/spl 0,10 dB/spl
Connector	μ_{con} σ_{con}	0,5 dB/con 0,2 dB/con
Additional coupling loss	μ_{cl} σ_{cl}	0 dB 0 dB

The maximum lengths estimated by iteratively solving equation (D.1) are found in table E.4.

Table E.4 - Calculated max. multimode length				
FC-0	LB	N_{con}	Length	Note
100-M5-SL-I	6	4	500 m	a.
100-M6-SL-I	6	4	175 m	b.
50-M5-SL-I	8	4	1,0 km	c.
50-M6-SL-I	8	4	350 m	d.
25-M5-SL-I	12	4	2,0 km	
25-M6-SL-I	12	4	700 m	e.
25-M5-LE-I				g.
25-M6-LE-I	6	4	1,5 km	f.
12-M5-LE-I				g.
12-M6-LE-I	6	4	1,5 km	

Notes:

a) This length limit of 500m is also the same length limit as imposed by the fiber bandwidth constraints (see table 18).

b) This length limit of 175m is due to fibre bandwidth contraints (see table 15). The length limit due loss budget (LB) would be 450m.

c) This length limit of 1,0 km is also the same length limit as imposed by the fiber bandwidth constraints (see table 18).

d) This length limit of 350 m is due to fibre bandwidth constraints (see table 15). The length limit due to loss budget (LB) would be 0,9 km.

e) This length limit of 700 m is due to fibre bandwidth constraints (see table 15). The length limit due to loss budget (LB) would be 1,8 km.

f) This length limit of 1,5 km is due to fibre bandwidth constraints (see table 15). The length limit due to loss budget (LB) would be 1,9 km.

g) For these cases the distance must be determined on a case by case basis.

Annex F
(informative)
Electrical cable interface implementation examples

F.1 Example of video coax cable characteristics

This large diameter style of video coax is capable of relatively long distance transmission due to its low attenuation and dispersion.

Table F.1 - Typical characteristics of 75 Ω, video coax (e.g.RG-6/U type)

Category Electrical	Impedance 75±5 Ω	Capacitance 5,2 pF/m nom.	Attenuation 11dB/50m @531MHz	Velocity 82%
Category Conductor	Material Bare Copper	Size AWG 18	Construction Solid	Outer diameter 0,96 mm nom.
Category Dielectric	Material Foamed polyethylene	Wall thickness 1,77 mm nom.	Dielectric Constant 1,50 nom.	Outer diameter 4,57 mm nom.
Category Shield	Material Tin plated Cu braid over foil		Coverage 100% min.	Outer diameter 5,84 mm nom.
Category Jacket	Material PVC	Wall thickness 0,51 mm nom.	Colour ——	Outer diameter 6,86 mm nom.

F.2 Example of plenum rated video coaxial cable characteristics

Table F.2 - Typical plenum rated video coaxial cable characteristics (e.g. RG-6/U type)

Category Electrical	Impedance 75±5 Ω	Capacitance 5,0 pF/m nom.	Attenuation 11dB/50m @531MHz	Velocity 82%
Category Conductor	Material Bare copper covered steel	Size AWG 18	Construction Solid	Outer diameter 1,01 mm nom.
Category Dielectric	Material Foamed Teflon	Wall thickness 1,65 mm nom.	Dielectric Constant 1,50 nom.	Outer diameter 4,37 mm nom.
Category Shield	Material Tin plated Cu braid over foil		Coverage 100% min.	Outer diameter 5,59 mm nom.
Category Jacket	Material Teflon	Wall thickness 0,203 mm nom.	Colour ——	Outer diameter 6,00 mm nom.

F.3 Example f miniature c ax cable characteristics

The attenuation of the miniature coax at 531 MHz is larger than the video coax characteristics specified in tables F.1 and F.2. The miniature coax, by virtue of its smaller physical size, is more lossy to high frequency harmonics of the signal than the video style cable.

Table F.3 - Typical characteristics of 75 Ω, miniature coax (e.g. RG-179 B/U type)

Category Electrical	Impedance 75±5 Ω	Capacitance 6,1 pF/m nom.	Attenuation 7dB/10m @531MHz	Velocity 70%
Category Conductor	Material Silver coated copper covered steel	Size AWG 30	Construction 7-38 stranded	Outer diameter 0,305 mm nom.
Category Dielectric	Material Extruded TFE Teflon	Wall thickness 0,635 mm	Dielectric Constant 2,10 nom.	Outer diameter 1,60 mm nom.
Category Shield	Material Silver coated Cu braid		Coverage 95% min	Outer diameter 2,11 mm nom.
Category Jacket	Material FEP	Wall thickness 0,203 mm nom.	Colour ———	Outer diameter 2,54 mm nom.

F.4 Example of double-shielded miniature coax cable characteristics

Table F.4 - Typical characteristics of 75 Ω, double-shielded miniature coax (e.g. miniature RG-59/U type)

Category Electrical	Impedance 75±5 Ω	Capacitance 5,2 pF/m nom.	Attenuation 7dB/10m @531MHz	Velocity 78%
Category Conductor	Material Copper covered steel	Size AWG 26	Construction Solid	Outer diameter 0,508 mm nom.
Category Dielectric	Material Foamed polyethelene	Wall thickness 0,711 mm	Dielectric Constant 1,50 nom.	Outer diameter 1,90 mm nom.
Category Shield #1	Material Bonded Alum. Tape		Coverage 96% min	Outer diameter 2,59 mm nom.
Category Shield #2	Material Aluminum braid		Coverage 60% min	Outer diameter 2,00 mm nom.
Category Jacket	Material PVC	Wall thickness 0,355 mm nom.	Colour ———	Outer diameter 3,71 mm nom.

F.5 Example of STP cable characteristics

This cable should be compatible with existing "Type 1" and "Type 2" 150 Ω STP cable as defined in EIA/TIA 568. The loss characteristics are shown in table F.5.

Table F.5 - Char of 150Ω, STP cable		
Half Baud Rate MHz	**Impedance Ohms**	**Attenuation dB/m (typ)**
66,406 25	150±15	0,092
132,812 5	150±15	0,138

F.6 Examples of coaxial data links

The coaxial data links may be transformer or capacitively coupled to provide common mode noise rejection and ground isolation. An example of a transformer coupled data link is shown in figure F.1.

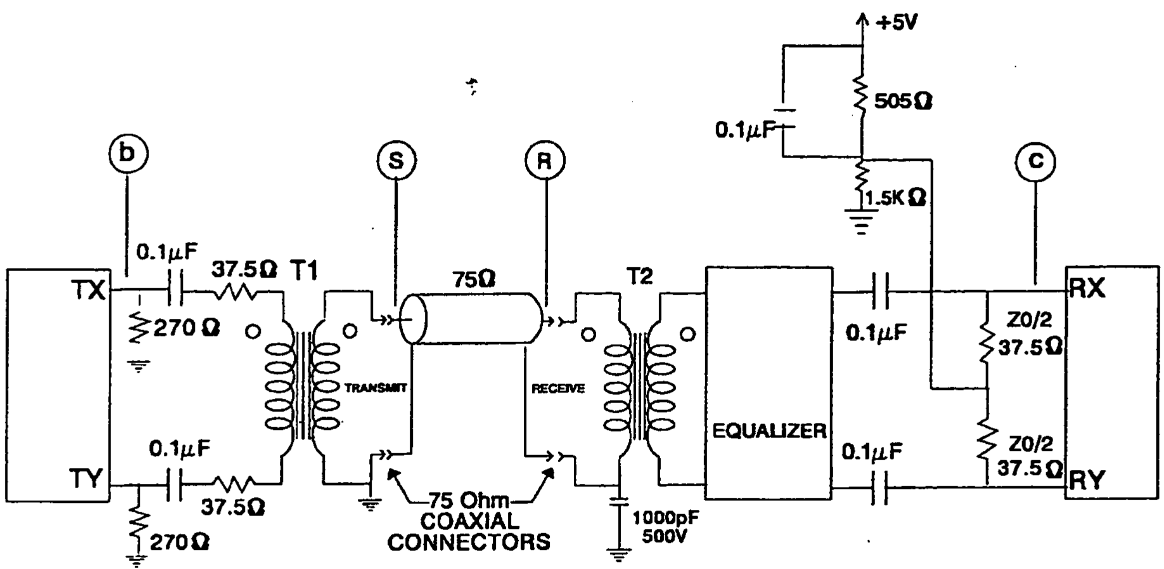


Figure F.1 - FC-0 with 75 Ω coaxial cable and transformer coupling

An example of a capacitively coupled data link is shown in figure F.2.

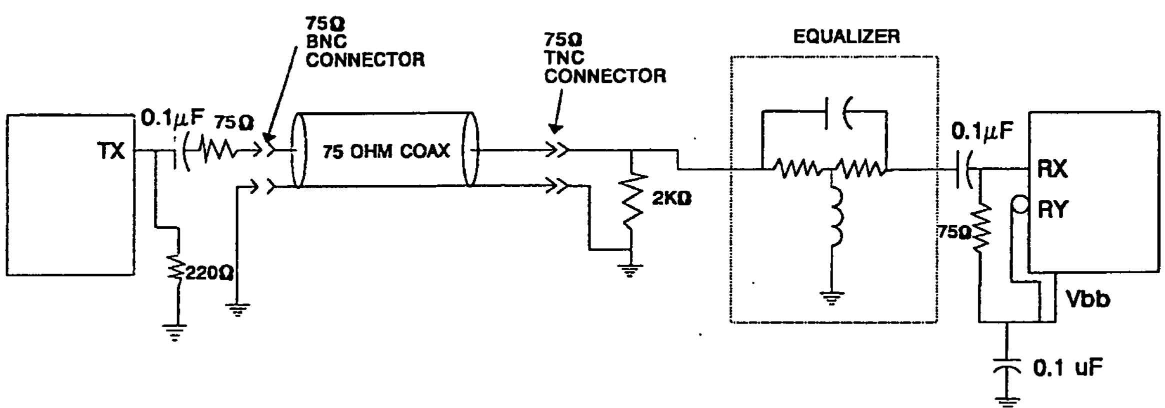


Figure F.2 - Example of coaxial cabl interface logic

F.7 Transf rmer characteristics

Transformer characteristics suitable for STP and coax applications are shown in table F.6.

Table F.6 - Characteristics of the Transformer

Baud rate MHz	Bandwidth (-3dB points) MHz	L µHy (min)		Rise/ Fall Time ns (max)	Ratio ±5%
		Coax	STP		
132,812 5	5 to 125	30	60	1,6	1:1
265,625	10 to 250	15	30	0,8	1:1
531,25	20 to 500	7,5		0,4	1:1
1 062,5	40 to 1 000	3,75		0,2	1:1

F.8 Terminator and optional equalizer network (description and design example)

The equalizer network shown by block diagram in figures 27 and 30 corrects for some of the signal distortion caused by the difference in amplitude loss and propagation delay time between frequency components of the transmitted signal. The primary physical loss mechanism is the skin effect characteristics of the cable. Except for these losses, the cable is extremely phase linear.

The equalizer network for the electrical data link should be designed to provide more attenuation at the lower frequencies and less attenuation at higher frequencies, thereby resulting in a flatter overall frequency response. At the same time, it should also exhibit either a linear phase characteristic(constant phase delay) or, preferably a phase delay characteristic with less delay at higher frequencies than at lower frequencies. The latter would tend to compensate for the phase delay characteristics of the cable. An example of an equalizer for a 100 MB/sec (531 MHz on the coax) link is shown in figure F.3 and table F.7. This particular equalizer achieves the two design objectives outlined by above. Also due to its design, it exhibits a considerable return loss at all frequencies, desirable for attenuating signal reflections. Attempting to compensate for the third- or fifth-order harmonics of the digital signal into the coaxial cable is usually not profitable in that it results in a rather elaborate and costly equalizer.

Note that the equalizer requirements for any link are greatly dependent on the particular coax cable employed and its length and condition, as well as any frequency-dependent attenuation and delay present in the transmitter and receiver circuits. The signal transformers, especially, can have a considerable effect on equalization requirements.

For lower data rates and/or shorter links, one may make the economic trade-off of elimination of the equalizer altogether at modest risk to error rate performance. At higher data rates, > 50 MB/sec (266 MHz on the Coax), an equalizer is recommended unless the cable length is short.

Table F.7 - Equalizer design example

Frequency	Attenuation	Delay
MHz	dB	ns
50	3,87 0	0,376
100	3,21 1	0,291
150	2,62 1	0,220
200	2,16 8	0,169
250	1,80 9	0,134
300	1,50 6	0,109
350	1,28 9	0,089
400	1,09 5	0,075
450	0,94 4	0,063
500	0,81 1	0,054
550	0,70 5	0,047
600	0,62 5	0,041
650	0,54 9	0,036
700	0,48 2	0,031
750	0,43 5	0,028
800	0,39 1	0,025
850	0,35 8	0,022
900	0,32 6	0,020
950	0,29 6	0,018
1 000	0,26 1	0,017

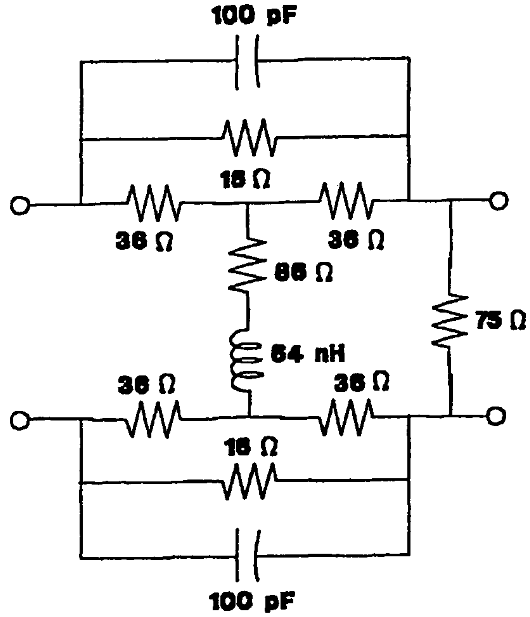


Figure F.3 - Equalizer design example

F.9 Example of STP data link

A typical example of an STP data link is shown in figure F.4.

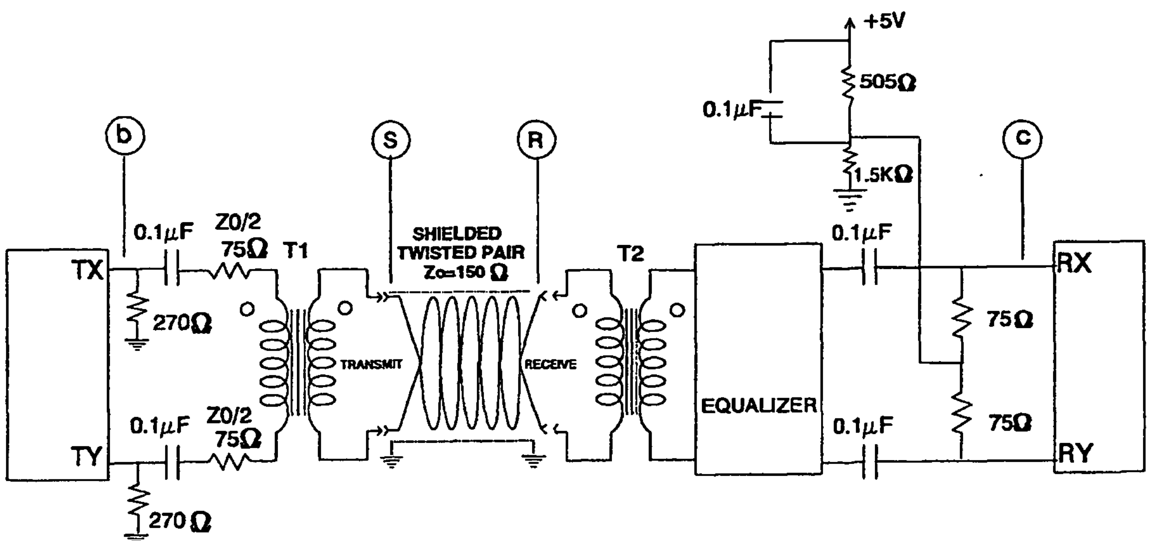


Figure F.4 - The STP link including an equalizer

Annex G
(informative)
Connector

G.1 Combined Connector Mechanical-Optical Requirements

Table G.1 - Plug/Receptacle Mechanical-Optical Requirements.
S/I = Standard vs. Informative

Tests	Receptacle			Plug		
	Value	Recommended test method	S/I	Value	Recommended test method	S/I
Axial pull force (latch retention force)	90N min.	Annex G.3	I	90N min. 1,0dB max var.	Annex G.7	I
Insertion/ withdrawal force	10N min. 80N max.	Annex G.4	I	80N max.	Annex G.8	I
Single plug repeat-ability	2,0dB max. (SM) 1,5dB max. (MM)	Annex G.5	I	Not Appli-cable	-	-
Cross plug repeatability	3,0dB max. (SM) 2,0 max. (MM)	Annex G.6	I	Not Appli-cable	-	-
Off axis (rotational) pull	Not Appli-cable	-	-	20N min. 1,0dB max. var.	Annex G.9	I
Cable/connector pull strength (cable to con-nector retention)	Not Appli-cable	-	-	90N min.	Annex G.10	I
Insertion/withdrawals	250 min.		I	250 min.		I

G.2 Connect r testing definiti ns and c nditi ns

The term "failure" shall be defined as

a) Increase in attenuation of 1,0 dB or a decrease in return loss of 5 dB as specified per test.

b) Mechanical damage defined as splitting, cracking, pitting, galling or deformation observable under 10X magnification unless otherwise stated for the Ferrule, Cable or Connector Body.

c) Functional dimensions non-conforming to print requirements after stress.

The application of a stimulus such as a force or environmental condition shall be applied once to demonstrate compliance with the requirement unless otherwise stated. The following tolerances shall apply to all connector tests:

Table G.2 - Connector test tolerances

Length	± 10%
Angles	± 3°
Rates	± 10%
Weights	± 10%

Before each of the following tests are conducted, clean the ends of the fibre optic connectors.

Unless otherwise stated, the Duplex Coupler shall be mounted between two frame members via the mounting slots.

When Duplex Cable Assemblies are inserted in the receptacle, they shall be cantilevered free of any support.

G.3 Receptacle axial pull test

Purpose: To ensure the receptacle is able to withstand an axial applied plug pull load without degrading optical performance.

Test method:
FOTP-6 is the recommended test method. The plug must be completely latched before the start of testing.

G.4 R ceptacle insertion/withdrawal forc t st

Purpose: To ensure the receptacle is designed to allow for easy insertion of the plug. The test would check the stiffness of the SC clips and ferrule/bore fit.

Test method:
FOTP-187 is the recommended test method.

G.5 Receptacle optical repeatability test

Purpose: Fibre to optical source alignment measurement.

Test method:
FOTP-34 is the recommended test method for receptacles.

A minimum of 25 single plug insertions should be repeated on a single receptacle or coupler. The maximum variation between any two insertions should meet the requirements stated in Table G.1.

G.6 Receptacle optical cross plug repeatability test

Purpose: Fibre to optical source alignment measurement.

Test method:
FOTP-34 is the recommended test method for receptacles.

A minimum of 25 different plug insertions should be performed on a single receptacle. The maximum variation between any two insertions should meet the requirements stated in Table G.1.

G.7 Plug axial pull test

Purpose: To ensure minimum plug to receptacle retention force with an axial applied load to the cable. Also tests cable's strain relief. This test is not a breakage test, cable functionality is required during the test.

Test meth d:
FOTP-6 is th recommended test method. The plug must be completely latched before the start of testing.

G.8 Plug insertion/withdrawal force test

Purpose: The true measure of ferrule float and housing to housing compliance. Testing would require gages to check the plug's ability to insert into bores at the four corners of the receptacle tolerance limits.

Test method:
FOTP-187 is the recommended test method. The couplers/adapters used for this evaluation should ideally represent the high end of geometric tolerance range in terms of bore position and alignment.

G.9 Plug off axis pull t st

Purpose: Ensures that cables can be dress in any direction without loss of optical performance.

Test method:
IEC 874-1, subclause 28.7.4 is the recommended test method.

G.10 Cable/ plug pull strength

Purpose: Test integrity of the fibre to connector strain relief.

Test method:

FOTP-6 is the recommended test method. The plug must be secured before the start of testing.

Annex H
(informative)
FC-0 service interface

This annex defines the interfaces of the communications media controls and services that are valid for all FC-0 data links. The controls and services are described in terms of logical primitives rather than physical signals to allow for the widest range of physical implementations.

The interface between FC-0 and FC-1 is intentionally structured to be technology and implementation transparent. That is, the same set of commands and services may be used for all signal sources and communications media. The intent is to allow for the the interface hardware to be interchangeable at the system level without regard to the technology of a particular implementation. As a result of this, all safety or other operational considerations which may be required for a specific communications technology are to be handled by the FC-0 level associated with that technology. Such safety features are provided by by the Link Control block of figure H.1.

H.1 General Description

As an aid in visualizing the FC-0 services refer to figure H.1. In this figure the general functions performed by the FC-0 level are illustrated along with the logical control associated with the functions. This diagram is meant to represent the most complex implementation of the FC-0 level. In some cases individual implementation, or some types of media, may not require all of these functions or logical communications.

For example, LED implementations will not require the FC-0_Signal_Detect to be communicated to the link control function. In the case of a SAW filter type of clock recovery the FC-0_Resync and the FC-0_Clock_Reference will not be required. The basic functions of the primitive controls and services are:

FC-0_Data.Request
: The serial data to be transmitted over the communications media.

FC-0_Data.Indication
: The retimed serial data stream received from the communications media.

FC-0_Transmit
: The command to turn the transmitter on and off.

FC-0_Transmit_State
: The present internal state of the transmitter.

FC-0_Signal_Detect
: An indication of the presence or absence of a signal on the communications media.

FC-0_Clock_Out
: The timing clock recovered from the incoming data.

FC-0_Clock_Reference
: A frequency reference used as an aid in acquiring bit synchronization.

FC-0_Resync
: A command from the FC-1 level to attempt to reacquire bit synchronization.

H.2 FC-0 States

H.2.1 Transmitter States

The transmitter is controlled by the FC-1 level. Its function is to convert the serial data received from the FC-1 level into the proper signal types associated with the operating media.

- **Transmitter Not-Enabled State**: A not-enabled state is defined as the optical output off for optical communication media and a logical zero for electrical media. This is the state of the transmitter at the completion of the power on sequence unless the transmitter is specifically directed otherwise by the FC-1 level.
- **Transmitter Enabl d State**: The transmitter shall be deemed to be in an enabled state when the transmitter is capable of operation within its specifications while sending valid data patterns.

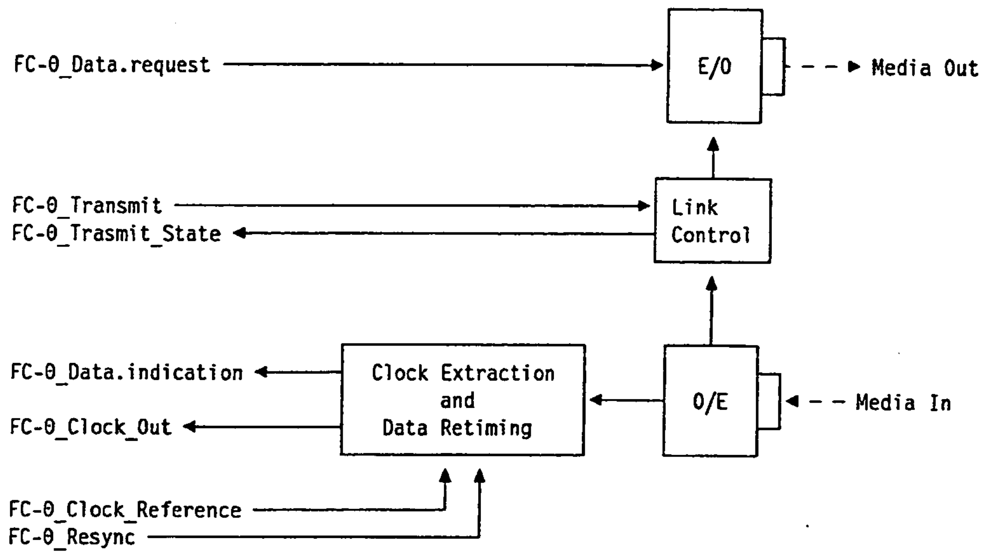


Figure H.1 - FC-0 logical structure

- **Transition Between Not-Enabled and Enabled States**: The sequence of events required for the transition between the not-enabled and enabled states are media dependant, both as to the time period required and the optical or logical activity on the media interface.
- **Transmitter Failure State**: Some types of transmitters are capable of monitoring themselves for internal failures. Examples are laser transmitters where the monitor diode current may be compared against a reference to determine proper operating point. Other transmitters, such as LEDs and electrical transmitters do not typically have this capability. If the transmitter is capable of performing this monitoring function then a detection of a failure should cause entry into the failure state.

H.2.2 Receiver States

The function of the receiver interface is to convert the incoming data from the form required by the communications media employed, retime the data, and present the data and an associated clock to the FC-1 level.

The receiver has no states.

H.3 FC-1 Services

Figure H.2 graphically portrays the Open Systems Interface (OSI) model as applied to the FC-0 service interface.

Primitives are of four generic types:

Request The request primitive is passed from the FC-1 level to the FC-0 level to request that a service be initiated.

Indication The indication primitive is passed from the FC-0 level to the FC-1 level to indicate an internal FC-0 event which is significant to the FC-1 level. This event may be logically related to a remote service request, or may be caused by an event internal to the FC-0 level.

Response The response primitive is passed from the FC-1 level to the FC-0 level to complete a procedure previously invoked by an indication primitive.

Confirm The confirm primitive is passed from the FC-0 level to the FC-1 level to convey the results of one or more associated previous service request(s).

Figure H.3 illustrates the primitive combinations that are supported by the FC-0 service interface. Each primitive described in this annex is correlated to one of the subfigures (a), (b), (c), or (d) of figure H.3. This figure also indicates the logical relationships of the primitive types. Primitive types that occur earlier in time and are connected by dotted lines in the diagrams are the logical antecedents of subsequent primitive types.

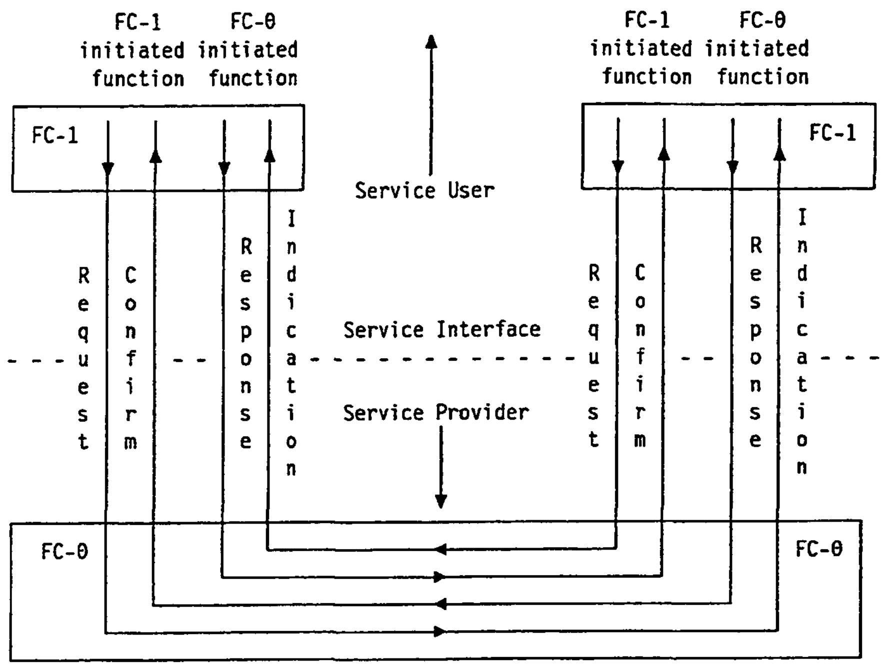


Figure H.2 - FC-0 service interface

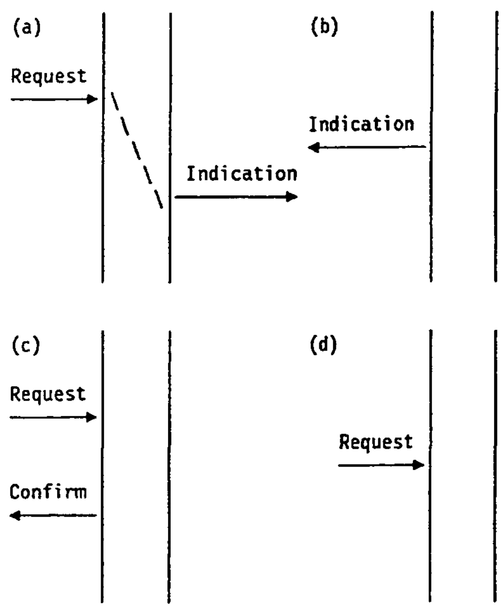


Figure H.3 - FC-0 service interface primitive combinations

The following service primitives are defined by the FC-0 service interface:

- FC-0_Transmit
  - request
- FC-0_Transmit_State
  - indication
- FC-0_Data
  - request
  - Indication
- FC-0_Clock_Out
  - indication
- FC-0_Clock_Reference
  - request
- FC-0_Resync
  - request
- FC-0_Signal_Detect
  - indication

H.4 FC-0_transmit.request

H.4.1 Function

This primitive is used by the FC-1 level to control the operational state of the FC-0 transmitter.

H.4.2 Semantics

FC-0_transmit.request(state)

state The requested state of the FC-0 transmitter. The allowable states of this parameter are enable, and disable. The format of this parameter is not defined by this standard.

H.4.3 When Generated

The FC-1 level issues the FC-0_transmit.request with state = enable to request that the transmitter begins its transition to an enabled state. During this time the FC-1 level must be supplying valid encoded information to the FC-0 level via the FC-0_data.request primitive.

The FC-1 level issues the FC-0_transmit.request with state = disable to request that the transmitter begin a transition to a not-enabled state.

H.4.4 Effect on Receipt

The receipt of this primitive with state = enable causes the FC-0 level to attempt to go to the enabled state. If the FC-0 level is in the not-enabled State the Transition state is entered. If the FC-0 level is in the Transition State the state is not affected. If the FC-0 level is in the Failure State no action is taken.

The receipt of this primitive with state = disable causes the FC-0 level to go to the not-enabled state from all states except the failure state. If the FC-0 level is in the failure state no action is taken.

H.4.5 Additional Comments

It is anticipated that for FC-0 media that do not have safety requirements the transition state will be a null state. For these cases the FC-0 level will go directly from the not-enabled state to the Enabled State.

When this primitive is issued with state=enable the FC-1 level must issue FC-0_data.request primitives continuously until this primitive is again issued with state=disable.

This primitive is illustrated by figure H.3 (d).

H.5 FC-0_transmit_state.indication

H.5.1 Function

This primitive is used by the FC-0 level to indicate to the FC-1 level that the transmitter is active and capable of transmitting data on the media interface.

H.5.2 Semantics

FC-0_transmit_state.indication(status)

status The state of the FC-0 transmitter. The allowable states of this parameter are active, inactive, and fault. The format of this parameter is not defined by this standard.

H.5.3 When Generated

The FC-0 level issues the FC-0_transmit_state.indication primitive with status = active continuously while the transmitter is in the enabled state and is capable of transmitting data on the media interface.

The FC-0 level issues the FC-0_transmit_state.indication primitive with status=fault when the transmitter is in an internal fault state as determined by any internal fault monitoring which it may possess. If the transmitter contains no internal fault detection circuits this is a null state.

The FC-0 level issues the FC-0_transmit_state.indication primitive with status = inactive continuously when the transmitter is not in the enabled or failure states. This will occur when the transmitter is in any of the following states:

- Transmitter not-enabled;
- Transmitter in transition between enabled and not-enabled states in either direction; or
- Transmitter not operational due to an internal fault which is not detected by internal fault monitoring circuitry.

H.5.4 Effect on Receipt

Upon receipt of this primitive with the status = active the FC-1 level may assume that any issued FC-0_data.request primitives will be placed on the communications media.

Upon receipt of this primitive with the status = inactive the FC-1 level will be informed that the response of the FC-0 level to any FC-0_data.request primitives is undefined.

Upon receipt of this primitive with the status = fault the FC-1 level will be informed that a repair activity is required.

H.5.5 Additional Comments

This primitive is illustrated by figure H.3 (b).

H.6 FC-0_data.request

H.6.1 Function

This primitive is used by the FC-1 level to pass transmission requests to an associated FC-0 level.

H.6.2 Semantics

FC-0_data.request(bit)

bit The information which is to be transmitted on the media. The allowable values of this parameter are one and zero.

H.6.3 When Generated

This primitive is generated by the FC-1 level. It is the serialized form of the data to be transmitted as formatted and coded by the FC-1 level.

H.6.4 Effect on Receipt

Upon receipt of this primitive the FC-0 level will change the communications media to the state appropriate to the parameter value, provided the FC-0 transmitter is in the enabled state as indicated by the FC-0_transmit_state.indication primitive with status = active. The response to this parameter will be undefined if the FC-0 level is not in the enabled state as indicated by the FC-0_transmit_state.indication primitive with status = active.

bit = one represents an optical high power state in the case of optical media and a logical one in the case of electrical media. bit = zero represents an optical low power state in the case of optical media and a logical zero in the case of electrical media.

H.6.5 Additional Comments

This primitive should be presented at a rate appropriate to the media in use. It must be presented continuously when FC-1 level has requested the transmitter to be in the enabled state by issuing the FC-0_transmit.request primitive with state=enable. Issuing of this primitive is optional when the FC-0_transmit.request primitive has been issued with state=disable.

This primitive is illustrated by the Request portion of figure H.3 (a).

H.7 FC-0_data.indication

H.7.1 Function

This primitive is used by the FC-0 level to pass data received from the media to the associated FC-1 level.

H.7.2 Semantics

FC-0_data.indication(bit)

bit The information which was received from the media. The allowable values of this parameter are one and zero.

H.7.3 When Generated

This primitive is generated synchronously with the FC-0_clock_out primitive. When the FC-0_clock_out.indication is issued the FC-1 level may receive the FC-0_data.indication primitive. When the FC-0_clock_out.indication is not present the value of the FC-0_data.indication is not guaranteed and the FC-1 level is advised to ignore the FC-0_data.indication primitive.

The parameter value of bit = one is generated by a high level of optical power in the case of

optical media or a logical one in the case of electrical media.

The parameter value of bit = zero is generated by a low level of optical power in the case of an optical media or a logical zero in the case of electrical media.

H.7.4 Effect on Receipt

The receipt of an FC-0_data.indication primitive by the FC-1 level allows the FC-1 level to receive data.

H.7.5 Additional Comments

In the event that there is no input to the FC-0 from the media this output is undefined.

This primitive is illustrated by the Indication portion of figure H.3 (a).

H.8 FC-0_clock_out.indication

H.8.1 Function

This primitive is used by the FC-0 level to provide synchronization information for the FC-0_data.indication to the FC-1 level.

H.8.2 Semantics

FC-0_clock_out.indication

No parameters are defined for this primitive.

H.8.3 When Generated

An FC-0 level issues the FC-0_clock_out.indication primitive to provide a data transmission synchronization signal to the FC-1 level. When this primitive is issued the FC-0_data.indication primitive will be in a valid state. At this time the FC-1 level may choose to receive FC-0_data.indication. At times between issuing the FC-0_clock_out.indication the FC-0_data.indication may not be in a valid state and it is recommended that the FC-1 level not sample the FC-0_data.indication primitive at these times.

H.8.4 Effect n Receipt

Receipt of this primitive may be used by the FC-1 level to synchronize data transmission between the FC-0 and FC-1 levels.

H.8.5 Additional Comments

In the event that there is no input to the FC-0 from the media the output frequency and timing information is undefined.

This primitive is illustrated by figure H.3 (b).

H.9 FC-0_clock_reference.request

H.9.1 Function

This primitive is issued by the FC-1 level as a aid in initializing the clock recovery functions of the FC-0 level.

H.9.2 Semantics

FC-0_clock_reference.indication(boundary)

boundary An indication of the transmission intervals of the reference clock. The allowable values of boundary are word and null.

H.9.3 When Generated

The FC-1 level issues this primitive with value boundary = word to indicate the reference times of the clock signal. This primitive has a value of boundary = null at intervals between these reference times. The FC-1 level must issue this primitive with boundary = word continuously at a rate appropriate to the physical implementation.

H.9.4 Effect on Receipt

This primitive is used in the FC-0 bit synchronization to initialize the clock recovery to near the bit frequency if a phase lock loop (PLL) form of clock recovery is used.

H.9.5 Additi nal C mments

In the event that the FC-0 implements the clock recovery using a method (e.g. SAW filters) which does not require a frequency reference this primitive will not be required.

This primitive is illustrated by figure H.3 (d).

H.10 FC-0_signal_detect.indication

H.10.1 Function

This primitive is issued by the FC-0 level to inform the FC-1 level of the presence or absence of a signal on the communications media.

H.10.2 Semantics

FC-0_signal_detect.indication(state)

state The result of the signal detect circuit. The allowable values of this parameter are present and absent.

H.10.3 When Generated

The FC-0_signal_detect.indication will have state = present when the receiver has determined that the signal on the input is above a predetermined level. In the case of optical media the activation level will lie in a range whose upper bound is the minimum specified sensitivity of the receiver and whose lower bound is the lower of either the point where the bit error rate will exceed 10^{-2} or 7 dB below the minimum receiver sensitivity. If the signal is below this range the primitive will be issued with state = absent.

Implementation of a signal detect function is optional. In the event that signal detect is not implemented the FC-0 shall issue this primitive with state = present.

While there is no defined hysteresis for this primitive the primitive shall undergo a single transition between the values of the state parameter for any monotonic increase or decrease in the optical power. The optical power at the transition points shall lie within the previously defined bounds.

For optical links that employ a link control function, such as those in annex I, then this "Signal Detect" is replaced by the "Link Status" signal.

In the case of optical media the reaction time of this primitive to a change in the input optical power is less than 12 s.

In the case of electrical media this primitive is issued with state = absent within 12 s of the occurrence of the fault. Otherwise State=Present will be issued.

H.10.4 Effect on Receipt

The FC-1 level uses this primitive to determine if a receiver which has been in a loss-of-synchronization state for more than one second should enter the loss-of-synchronization-failure state or the loss-of-signal-failure state.

H.10.5 Additional Comments

This primitive is illustrated by figure H.3 (b).

H.11 FC-0_resync.request

H.11.1 Function

This primitive is used by the FC-1 level to request that the FC-0 attempt to acquire bit synchronization.

Some bit synchronization methods do not require an initialization procedure. In the event that one of these methods is employed by the FC-0 level this primitive is not required.

H.11.2 Semantics

FC-0_resync.request

No parameters are defined for the FC-0_resync.request primitive.

H.11.3 When Generated

This primitive is issued by the FC-1 level to request that the FC-0 level attempt to acquire bit synchronization. This may occur as a result of an FC-1 reset condition.

H.11.4 Effect on R ceipt

When this primitive is received the FC-0 level will begin any bit synchronization procedure appropriate to the specific implementation. The method of acquiring bit synchronization is not a subject of this standard.

The synchronization procedure should take less than 1 ms from the receipt of this primitive.

H.11.5 Additional C mments

The time required to achieve bit synchronization is highly variable depending on the particular bit synchronization method employed.

This primitive is illustrated by figure H.3 (d).

Annex I
(informative)
The open fibre control interface

The Open Fibre Control (OFC) safety interlock system, four-state machine, and handshake timings are specified in 6.2.3; much of the material is repeated here for completeness. This annex presents a recommended implementation and a justification of the maximum power and timing values specified in 6.2.3.

I.1 OFC Interface Description

I.1.1 System overview

The OFC system functions as a safety interlock for a point-to-point optical data link by detecting whenever the link is disrupted (e.g., cut fibre or disconnected connector) and forcing the transceivers into a repetitive pulsing mode of operation with very low duty cycle. The link can return to normal data traffic only after the OFC system detects that the link has been repaired and the proper reconnection handshake has taken place between the two transceivers in the link.

I.1.1.1 Input lines

Input lines to the OFC system consist of a system clock and two independent loss-of-light (LOL) detector lines that indicate whether-or-not an optical signal is being received by the photodetector/receiver. At least one of the two LOL detectors is a.c. coupled to the photodetector/receiver and therefore sensitive only to modulated optical signals (see figure 25).

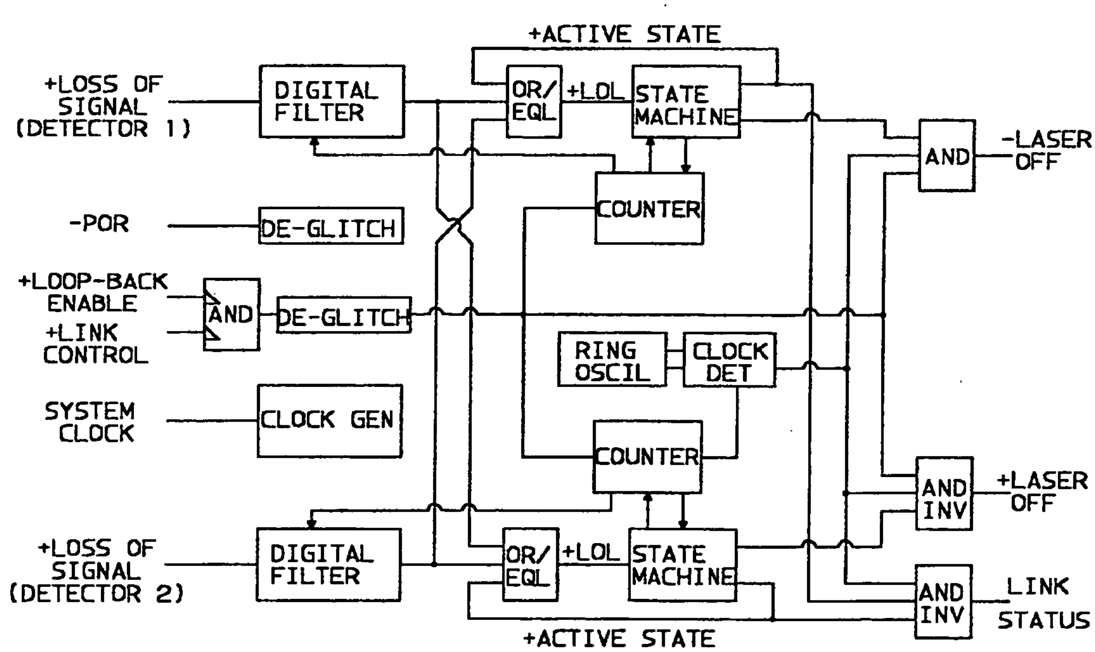


Figure I.1 - Possible impl mentation of the OFC system

I.1.1.2 Output lines

The output lines consist of two laser driver control lines and an optional link status line. The two laser driver control lines are of opposite polarity to prevent voltage control problems from accidentally activating the laser. They are also each independently capable of disabling the laser drive circuitry (via separate control paths). The link status line is used to signal the user system when the link is inactive due to a LOL condition detected by the OFC system.

I.1.1.3 Additional control lines

In addition to the input and output lines, there is a power-on-reset line and two optional user system control lines that interface with the OFC system. The power-on-reset line is used to synchronize the counters and state machines in the system. The two user system control lines, link control and loop-back enable, interact with the OFC system by forcing the OFC to disable the laser drive circuitry and turn off the laser. Once the laser is turned off, only the OFC system can turn it back on by performing a link reconnection handshake.

I.1.1.4 Maximum optical power and handshake timings

The repetitive pulsing and reconnection handshake are controlled by logic in two state machines contained in the OFC system. The two state machines are independent and identical. This redundancy found throughout the OFC system is implemented to satisfy a requirement found in most laser safety standards which states that the safety interlock system must remain functional during single fault conditions.

The OFC timings used during a reconnection attempt depend upon the maximum (worst case) average power accessible from a transmitter receptacle port, the circuit components and technology, and the restrictions imposed by laser safety standards. The OFC timings for the 1062 and 531 Mbaud links are different than the timings for the 266 Mbaud link due to the increase in allowed optical power for the 1062 and 531 Mbaud links versus that for the 266 Mbaud link.

a) Average transmitter receptacle power (maximum):
   - 25-M5-SL-I: 1,7 dBm (1,48 mW)
   - 50-M5-SL-I: 3,2 dBm (2,10 mW)
   - 100-M5-SL-I: 3,2 dBm (2,10 mW)

b) Pulse repetition time, T, (Disconnect State)
   - 25-M5-SL-I: 10,1 s (2^{28} T_{25})
   - 50-M5-SL-I: 10,1 s (2^{29} T_{50})
   - 100-M5-SL-I: 10,1 s (2^{30} T_{100})

c) Pulse duration, t, (decode 1 time period in Disconnect State and decode 3 time period in Reconnect State)
   - 25-M5-SL-I: 617 μs (2^{14} T_{25})
   - 50-M5-SL-I: 154 μs (2^{13} T_{50})
   - 100-M5-SL-I: 154 μs (2^{14} T_{100})

d) Stop time, (decode 2 time period in Stop State)
   - 25-M5-SL-I: 1 234 μs (2^{15} T_{25})
   - 50-M5-SL-I: 617 μs (2^{15} T_{50})
   - 100-M5-SL-I: 617 μs (2^{16} T_{100})

where T_{25}, T_{50} and T_{100} are the (10 bit) byte times for the respective links and the tolerance in the timings is that which arises from the bit rate tolerance.

While in the inactive mode of operation, the maximum time for the port to respond to a "light present" or "loss-of-light" signal at its receiver is limited by the pulse duration, since the worst case round trip time must be less than this value. This includes any delays from light detection, filtering, the state machine, laser turn on or off, and fibre transmission.

While in the active mode of operation (i.e., normal transmission), the turn-off time, T_{off}, specifies the maximum time between the instant that the data link is disrupted or a fault condition is detected to the instant that laser light is no longer emitted from any point of access in the data link (i.e., each port has switched to inactive mode operation). This turn-off time is longer than the pulse duration or stop times to allow for improved filtering and reliability of the loss-of-light signal. It is specified as follows:

Activ mode link turn-off time, T_{off}, (maximum):
- 25-M5-SL-I: 4.0 ms
- 50-M5-SL-I: 2.0 ms

– 100-M5-SL-I: 2.0 ms

I.1.2 Block Diagram description of the OFC System

A block diagram of a module that implements the OFC system control is shown in figure I.1. The discussion which follows describes the function of each of the blocks.

The OFC module contains two control paths that must be satisfied before the laser can be activated. The two paths provide the required redundancy. Each path contains a digital filter, state machine and a counter. The internal redundancy is complemented externally, by two LOL detectors and two control paths in the laser drive circuitry.

The two loss-of-light detectors each feed a digital filter; the output of each filter is OR/EQUAL'd to form an internal Loss-of-Light (LOL) signal. In the Disconnect, Stop and Reconnect States the "EQUAL" function is implemented. The internal LOL signal is allowed to toggle from asserted to deasserted or vice versa only when the two filter outputs agree. Once the system is in the Active State, the "OR" function is implemented so that only one light detector is required to assert LOL and deactivate the laser.

The digital filters integrate the incoming signals to improve their reliability. The digital filters sample at a faster rate during the reconnect sequence than during normal operation when asserting LOL and signalling a link disconnection.

The internal LOL signals are used to synchronize the counters and state machines. The counters control the pulse repetition, pulse duration and stop times. The counters also provide the low frequency sampling clock to the digital filters. The state machines control the handshake algorithm implemented in the OFC system and independent "Laser Off" output lines. Each "Laser Off" output line is independently capable of disabling the laser drive circuits via separate control paths.

The OFC module contains a ring oscillator. The ring oscillator drives a clock detector that monitors the 'System Clock' input signal. If the 'System Clock' is stuck high or low the clock detector causes the laser to be deactivated. This provides a back up safety feature to the single clock input signal. The 'System Clock' only needs to slow down sufficiently for the clock detector to detect a fault. If the 'System Clock' speeds up, all of the timings scale proportionately so that the ratio of the pulse duration time to the pulse repetition time remains constant.

The response of the module to the external control signals, +Loop Back Enable, +Link Control and -POR, must not cause a pulse to be emitted when they are toggled. The logic must ensure that a safe time period exists between pulses. Hence any disruption of the OFC system must cause the state machine to be reset to the Disconnect State.

I.1.3 The OFC State Machine

The OFC state machine contains the control system logic that detects when the optical link is opened due to a disconnection or break in the fibre and presides over the link reconnection handshake when it detects that the link is once again closed. A state diagram of the algorithm implemented in the OFC system is shown in figure I.2.

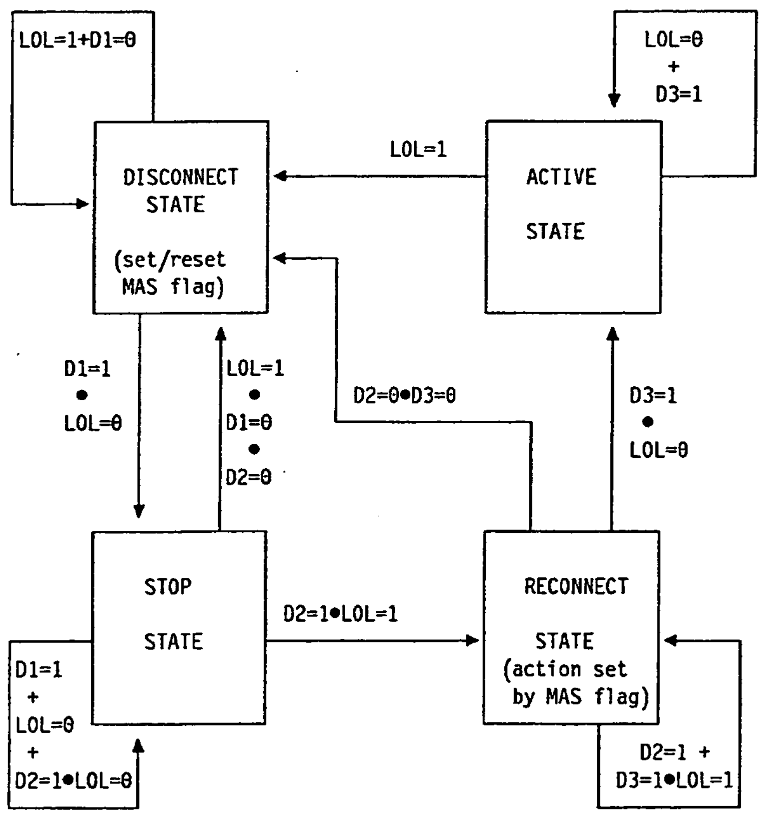
The state machine uses five flags to control the transitions between the states. The loss-of-light (LOL) flag is the "EQUAL" of the two light detector input lines while in the Disconnect, Stop and Reconnect States, and the "OR" of the two light detector lines while in the Active State. In other words, during link activation this flag is asserted and deasserted only when the signals from both light detectors agree, but once activated, either light detector sensing a "no light present" condition causes the LOL flag to be asserted. The "master of link reconnection" (MAS) flag is used to ensure that if a transceiver is the responding rather than the initiating port in a send/receive exchange while in the Disconnect State, then it must also be the responding and not the initiating port during the Reconnect State exchange. This flag prevents problems that could arise from timer variations and link synchronization. The three decodes are generated by the system clock and counter in the OFC system. The decodes are used to ensure that no ON-OFF-ON sequence generated by the physical insertion of a fibre into the connector can accidentally satisfy the safety algorithm.

The following list describes each of the four states of operation of the OFC system (figure I.2 contains a diagrammatic description, and the timing values where given in the I.1.1.4.)

a) Disconnect state:

The purpose of the Disconnect State is to prevent laser emissions from exceeding Safety Class 1 limits while the link is opened and to repetitively check for a closed link. The OFC system accomplishes both of these tasks by operating the port's laser at a very low duty cycle while in this state. The laser is activated for only t μs every T seconds to check for a closed optical link between itself and the transceiver at the opposite end of the link. As long as the port's LOL flag remains asserted, the OFC system keeps the transceiver in this state. To exit from the Disconnect State, light must be both sent and received by the transceiver. This send/receive exchange can occur in two ways:

1) If the T seconds timer expires before the transceiver receives an optical signal from the port at the opposite end of the link, then decode D1 is asserted (D1=1), the MAS flag is asserted (MAS=1), and the port's laser is activated for the duration of the decode 1 period (t μs). If during this decode 1 period an optical signal is received from the port at the opposite end of the link (i.e., LOL=0), then the port's OFC system proceeds to the Stop State for the remainder of the decode 1 period. The asserted MAS flag implies that this transceiver unit initiated the link reconnection



DEFINITIONS:
LOL = Loss-of-light flag (asserted = 1, deasserted = 0)
MAS = Master of link reconnection flag
D1 = Decode 1 - 1st time period flag = Link check
D2 = Decode 2 - 2nd time period flag = Disable laser
D3 = Decode 3 - 3rd time period flag = Link check

Figure I.2 - Open Fibre Control module Stat Diagram

check by sending light first and receiving light second; it is considered the "master" of the link reconnection.

2) If an optical signal is received (LOL=0) from the port at the opposite end of the link sometime during the T seconds wait period, then the counters controlling the timing are reset, decode D1 is asserted (D1=1), the MAS flag is deasserted (MAS=0), and the laser is activated for the duration of the decode 1 period (t μs). Since the send/receive exchange is complete, the port's OFC system proceeds to the Stop State. The deasserted MAS flag implies that this transceiver unit is responding to a link reconnection attempt from the unit at the opposite end of the link. By receiving light first and sending light second, it is considered the "slave" of the link reconnection attempt.

b) Stop state:

The Stop State is the "off" portion of the "on-off-on" reconnection algorithm. In the Stop State the OFC system forces the port's laser off. However, the laser is not disabled until after the decode 1 (D1) period is complete which ensures that both of the ports are transmitting long enough to satisfy the Disconnect State exit condition at each port. When the decode 1 period ends, D1 is deasserted (D1=0), D2 is asserted (D2=1) and the laser driver circuitry is disabled for the duration of the decode 2 period (typically ≥ 2t μs). The OFC system remains in the Stop State for as long as light is detected (i.e., LOL=0), conceivably for an indefinite period of time. There are two exit paths from the Stop State:

1) One exit from the Stop State is to continue on to the Reconnect State. This exit condition occurs when light is no longer detected (LOL=1) prior to the decode 2 period expiring. This exit path is the normal chain of events during a reconnect handshake.

2) The other exit from the Stop State takes the port's OFC system back to the Disconnect State. This exit path is used when the loss of the light signal (LOL=1) occurs after the decode 2 period has expired (i.e., D1=0 and D2=0). Since an optical signal remained present throughout the decode 2 period, the OFC system assumes that the other end of the link does not have a compatible safety control system and therefore rejects the link connection attempt.

c) Reconnect state:

The Reconnect State verifies that a closed link exists between the two ports in the link by once again requiring that an optical signal be both sent and received during a t microsecond check period (referred to as the decode 3 period). In this state the function of the OFC system is different for the master (MAS=1) and slave (MAS=0). If a transceiver unit responded to an optical signal in the Disconnect State (i.e., MAS=0), it is important that it also responds in the Reconnect State and does not attempt to initiate the send/receive exchange.

When the OFC system enters the Reconnect State, decode 2 is asserted (D2=1) and the laser is disabled. The OFC system continues to keep the laser disabled until the decode 2 time period expires, then one of the following sequences of events occur:

1) If the transceiver is master of the connection attempt (i.e., MAS=1), then decode 3 is asserted (D3=1) and the laser is activated for the duration of the decode 3 period (t μs). If during the decode 3 period an optical signal is received in response to this initiating light pulse (i.e., D3=1 and LOL=0), then the OFC system exits to the Active State. Otherwise, when the decode 3 time period ends (D2=0 and D3=0), the OFC system disables the laser and exits to the Disconnect State.

2) If the transceiver is slave of the connection attempt (i.e., MAS=0), then decode 3 is asserted (D3=1) but the laser is not activated. If during the decode 3 period (t μs) an initiating optical signal is received from the master (i.e., D3=1 and LOL=0), then the OFC system activates the laser in order to send a response and exits to the Active State. Otherwise, when the decode 3 period ends (D2=0 and D3=0), the OFC system continues to keep the laser disabled and exits to the Disconnect State.

d) Active state:

The Active State is for normal point-to-point data communications. In the Active State the OFC system allows the laser to function continuously while it monitors the two LOL detectors. The OFC system remains in the Active

State for as long as an optical signal is received by both of the detectors (i.e., LOL=0). If either of the detectors sense a LOL condition, the laser is disabled, the LOL flag is asserted (LOL=1) and the OFC system transfers control to the Disconnect State.

I.2 Laser Safety Standards and OFC Timing Specifications

I.2.1 Laser Safety Standards

Although there is a considerable amount of similarity between the laser safety standards and regulations that exist throughout the world, some requirements differ, especially with respect to labeling and certification. Within the U.S., all laser products must be certified by the manufacturer to conform to the requirements contained in the F.D.A. regulation 21 CFR subchapter J. In addition, it may be a business requirement to conform to the Z136.2 laser safety standard produced by the American National Standards Institute (ANSI). Outside of the U.S., many countries base their laser safety regulations on the International Electrotechnical Commission (IEC) 825 laser safety standard. The time values used for the OFC system in this Fibre Channel standard are based on the emission requirements for Safety Class 1 laser products contained in the IEC 825 standard (1984 plus amendment 1, 1990). The reason this standard was used over the other two is that the IEC emission requirements for a Safety Class 1 optical fibre system are more restrictive, and the goal is to specify an OFC system interface which satisfies worldwide Safety Class 1 emission requirements.

I.2.2 The OFC Timing Specifications

An optical fibre transmission system is a closed system (i.e., no accessible laser emissions) during normal link operating conditions and therefore a Safety Class 1 system. It is only during maintenance and service conditions, when the optical path is accidentally or purposefully broken, that access to laser emissions is possible. The point in the transmission link that has the largest emission level is at the transmitter receptacle of a transceiver. Classification is therefore based on the maximum emission level at the transmitter receptacle. This maximum value is a worst case value and includes variations due to temperature effects, lifetime effects and single faults.

The OFC system uses a repetitive pulsing technique (t μs on every T seconds) during the time that a link is open in order to reduce the maximum possible exposure to a value allowing for classification as a Safety Class 1 laser product. The maximum average power level per pulse is a function of the wavelength, pulse duration (t), and pulse repetition frequency (PRF = 1/T). The PRF determines the number of pulses (N) that occur during the time base used for classification. It is important to note that the use of the word "pulse" refers to the time (t) during which the laser is powered on and being modulated with a valid full rate data pattern. From a laser safety point of view, this "pulse" can be thought of as a CW pulse of duration t and power level equal to the average power of the modulated signal.

The OFC system described above contains a natural potential for a two pulse emission every T seconds when only one of the two linking fibres is disconnected. This condition occurs if the optical transmission path from port A to B is open, but the path from B to A is closed, and the T seconds timer on transceiver A is sufficiently faster than the T seconds timer on transceiver B. (see Figure 23) Transceiver A emits a t μs pulse when its T seconds timer expires in an attempt to link up as master; this attempt fails since the link is open. A short time later, a second t μseconds pulse is emitted by transceiver A in response to transceiver B sending it a link check pulse along the closed part of the link. Since the T second timer on transceiver A is faster than that on B, this pattern is repeated until the link is repaired. Hence the worst case scenario is one that includes two (t μs) pulses every T seconds and must be taken into consideration in any safety calculations. In addition, the maximum turn-off time, T_{off}, of the link, must be considered as part of the total exposure since this turn-off delay initiates the repetitive pulsing mechanism.

As an example of laser safety calculations for the OFC system, we show that the timing values used in the 531 Mbaud SW data link meet the Safety Class 1 emission requir ments contained in the IEC 825 (1984 plus amendment 1, 1990) laser safety standard. The accessible emission limit (AEL) is defined as the maximum accessible laser emission level for a particular classification. For the r petitively pulsed situation in

the OFC system, the AEL for the pulse train is defined as

$$AEL_{train} = AEL_{single} \times N^{-0,25}$$

where:

AEL_{train} = exposure from any single pulse in the train
AEL_{single} = AEL for a single pulse
N = number of pulses during the applicable time base

The Safety Class 1 AEL for a single pulse in the wavelength range 700 to 1050 nm and emission duration, t, between 1.8×10^{-5} and 1000 seconds is given by the equations:

$$\begin{aligned} AEL_{single} &= 7 \times 10^{-4}\, t^{0,75}\, C_4\ J \\ &= 0.7\, t^{-0,25}\, C_4\ mW \end{aligned}$$

and

$$\begin{aligned} C_4 &= 10^{(\lambda - 700)/500} \\ &= 1,38 \quad (\lambda = 770\, nm) \end{aligned}$$

The change from units of energy to units of power is used since a balanced data code and high data rates imply that the average power level is essentially constant during the emission duration. Using the 531 Mbaud pulse duration, t = 154 μs, the single pulse AEL is

$$AEL_{single} = 8,67\ mW \quad (t = 154\, \mu s).$$

For wavelengths greater than 400 nm and situations where intentional viewing is not inherent in the design of the product, the time base is 1 000 s. Thus the worst case number of pulses during the time base is

$$\begin{aligned} N &= \left(\frac{1000}{T}\right) \times 2 + \left(\frac{T_{off}}{t}\right) \\ &= \left(\frac{1000}{10.1}\right) \times 2 + \left(\frac{2000}{154}\right) = 211 \text{ pulses} \end{aligned}$$

where T is the pulse repetition time (T = 10,1 s), the two pulse potential is included, and the effect of T_{off} has been included to account for the initial laser turn-off time. The AEL for the train of pulses can now be calculated:

$$AEL_{train} = 8,67 \times (211)^{-0,25} = 2,28\ mW$$

Comparing this value to the maximum (i.e., worst case) average transmitter receptacle power, 2,10 mW, we find that the specified maximum value is less than the AEL for the train of pulses and allows for close to a 10% guard band. Thus, a 531 Mbaud SW laser transceiver that implements the OFC system as described in this standard should be classifiable as a Safety Class 1 laser system with respect to IEC 825 (1984 plus amendment 1, 1990).

Annex J
(informative)
System jitter allocations

This annex contains examples of jitter budgets for Fibre Channel links.

J.1 Jitter sources

Jitter in the fibre optic components consists of Deterministic Jitter (DJ) and Random Jitter (RJ). In a realistic system there are also components relating to the clock recovery process. This annex only discussed the jitter allocations up to the input of the clock recovery function. See annex A for additional information on jitter measurements and definitions.

Jitter values are expressed as peak-to-peak unit interval (UI) values. A unit interval is one baud interval at the data rate under consideration. This representation method has been chosen to facilitate comparison of the different data rates described by Fibre Channel.

The peak-to-peak values are defined as that value for which the probability that it will be exceeded is equal to 10^{-12}. The components of deterministic jitter are directly added to arrive at the total DJ. For a Gaussian probability of random jitter, the different components in the link are assumed to be uncorrelated and may be added as the square root of the sum of their squares. For the Gaussian probability assumption the peak-to-peak jitter is evaluated as 14 times the RMS jitter. The total jitter at any point is the arithmetic sum of the deterministic and random jitters at that point. It is known that the transition times of the signal in the media affects the jitter contribution of various components in a link. To facilitate analysis the rise and fall times, in units of nanoseconds, are listed in table J.1. The transition times for the electrical and LED variants are measured from 10% to 90% of the signal. For lasers the measurements are made at the 20% to 80% points when measured through a fourth order Bessel-Thompson filter. For more information on rise and fall time measurement, see A.1.2. This jitter budget is provided to document the thinking underlying the FC-0 specifications and serve as guidance for the development of physical components for FC-0 implementations. Conforming Fibre Channel ports are required to comply only with requirements expressed in the main body of the of this standard. These requirements are noted in boldface underlined type in table J.1. For the interconnection of conforming Fibre Channel attachments, the true requirement for interoperability is at the optical interface provided by each attachment. These requirements are given in clauses 6 and 7.

J.2 Jitter allocations by technology variant

In table J.1 the jitter allocations are given in unit intervals. The jitter locations are referenced to points b, R, S, and c of figures 9, 10, 27 and 30. These points represent the following locations in a practical realization of Fibre Channel

b	serial electrical data. This is the serial representation of the transmission characters. It occurs in at the output of the serializer prior to the being processed by the circuitry required for conversion to the form required by the transmission medium.
b to S	the transmitter. This converts the electrical data to the form required for transmission on the communications medium. For optical classes this is the electrical to optical converter. For coaxial classes this is the source terminating network.
S	the transmitter reference point in the transmission medium. This point is on the medium side of the output connector.
S to R	the transmission medium
R	the receiver reference point in the transmission medium. This point is on the medium side of the input connector.
R to c	the receiver. This converts data from the form required for the transmission medium to serial electrical data. For the optical classes this is the optical to electrical converter. For the coaxial classes this is the equalizer network.

c serial electrical data. This is the unretimed electrical data prior to retiming.

Table J.1 - Jitter allocations

Variant	t_r/t_f (ns)		Jitter (UI)							
	S	R	component	b	b to S	S	S to R	R	R to c	c
100-SM-LL-L	0,37	NA	DJ	0,08	0,12	0,20	0	0,20	0,08	0,28
			RJ	0,12	0,20	0,23	0	0,23	0,35	0,42
			Total	0,20	0,32	0,43	0	0,43	0,43	0,70
100-SM-LL-I	0,37	NA	DJ	0,08	0,12	0,20	0	0,20	0,08	0,28
			RJ	0,12	0,20	0,23	0	0,23	0,35	0,42
			Total	0,20	0,32	0,43	0	0,43	0,43	0,70
100-M5-SL-I	0,37	0,6	DJ	0,08	0,12	0,20	0,03	0,23	0,08	0,31
			RJ	0,12	0,20	0,23	0	0,23	0,31	0,39
			Total	0,20	0,32	0,43	0	0,46	0,39	0,70
100-TV-EL-S 100-MI-EL-S	0,4	0,7	DJ	0,08	0,02	0,10	0,31	0,41	-0,10	0,31
			RJ	0,12	0,01	0,12	NA	NA	NA	0,39
			Total	0,20	0,03	0.22	NA	NA	NA	0,70
50-SM-LL-L	0,75	NA	DJ	0,08	0,12	0,20	0	0,20	0,08	0,28
			RJ	0,12	0,15	0,19	0	0,19	0,37	0,42
			Total	0,20	0,27	0,39	0	0,39	0,45	0,70
50-M5-SL-I	0,75	1,0	DJ	0,08	0,12	0,20	0,03	0,23	0,08	0,31
			RJ	0,12	0,15	0,19	0	0,19	0,34	0,39
			Total	0,20	0,27	0,39	0,03	0,42	0,42	0,70
50-TV-EL-S 50-MI-EL-S	0,6	1,0	DJ	0,08	0,02	0,10	0,32	0,42	-0,11	0,31
			RJ	0,12	0,01	0,12	NA	NA	NA	0,39
			Total	0,20	0,03	0,22	NA	NA	NA	0,70
25-SM-LL-L	1,5	NA	DJ	0,08	0,12	0,20	0	0,20	0,08	0,28
			RJ	0,08	0,15	0,17	0	0,17	0,38	0,42
			Total	0,16	0,27	0,37	0	0,37	0,46	0,70
25-SM-LL-I	1,5	NA	DJ	0,08	0,12	0,20	0	0,20	0,08	0,28
			RJ	0,08	0,15	0,17	0	0,17	0,38	0,42
			Total	0,16	0,27	0,37	0	0,37	0,46	0,70
25-M5-SL-I	1,5	2,0	DJ	0,08	0,12	0,20	0,03	0,23	0,08	0,31
			RJ	0,08	0,15	0,17	0	0,17	0,35	0,39
			Total	0,16	0,27	0,37	0,03	0,40	0,43	0,70
25-M6-LE-I	2,0/2,2	2,5	DJ	0,08	0,08	0,16	0,03	0,19	0,09	0,28
			RJ	0,08	0,03	0,09	0	0,09	0,41	0,42
			Total	0,16	0,11	0,25	0,03	0,28	0,50	0,70
25-TV-EL-S 25-MI-EL-S 25-TP-EL-S	1,2	2,5	DJ	0,08	0,02	0,10	0,33	0,43	-0,12	0,31
			RJ	0,08	0,01	0,08	NA	NA	NA	0,39
			Total	0,16	0,03	0,18	NA	NA	NA	0,70
12-M6-LE-I	4,0	4,5	DJ	0,08	0,16	0,24	0	0,24	0,19	0,43
			RJ	0,08	0,09	0,12	0	0,12	0,35	0,37
			Total	0,16	0,25	0,36	0	0,36	0,54	0,80
12-TV-EL-S 12-MI-EL-S 12-TP-EL-S	1,8	3,6	DJ	0,08	0,02	0,10	0,34	0,44	0	0,44
			RJ	0,08	0,01	0,08	NA	NA	NA	0,36
			Total	0,16	0,03	0,18	NA	NA	NA	0,80

J.3 Jitt r allocation example

As an example of jitter allocation consider the 25-M6-LE-I variant of table J.1. As the jitter allocations are all given in unit intervals the first step is to determine the unit interval time for the specific variant. This is given by the reciprocal of the bit rate.

$$UI = \frac{1}{0{,}265\ 625\ \text{Gbaud}} = 3{,}765\ \text{ns} \qquad (I.1)$$

The only requirements for Fibre Channel compliance are noted by the boldface underlined entries in table J.1. From this we may determine the maximum jitter at the output of a node from the allocations at point S.

$$\begin{aligned} DJ_S &= 3{,}765 \times 0{,}16 &&= 0{,}60\ \text{ns} \\ RJ_S(PP) &= 3{,}765 \times 0{,}09 &&= 0{,}34\ \text{ns} \\ RJ_S(RMS) &= 3{,}765 \times 0.09/14 &&= 0{,}024\ \text{ns} \end{aligned} \qquad (I.2)$$

The jitter allocations at the input of the receiving node are give by the allocations of point R. The only difference between this point and the transmitter point is the 3% DJ contribution of the fibre. This results in a DJ input to the receiver of

$$DJ_R = 3{,}765 \times 0{,}19 = 0{,}72\ \text{ns} \qquad (I.3)$$

The RJ requirements of the receiver are the same as the transmitter requirements.

While the above jitter allocations are the only ones required for Fibre Channel compliance, table J.1 may be used to determine recommended design points for the components used to construct the link. The components of interest would be the transmitter and receiver modules and the chip set used to supply the serial data.

The transmitter and receiver module requirements would be given by the b to S and the R to c columns of table J.1. The results for the transmitter are:

$$\begin{aligned} DJ_{TX} &= 3{,}765 \times 0{,}08 &&= 0{,}30\ \text{ns} \\ RJ_{TX}(PP) &= 3{,}765 \times 0{,}03 &&= 0{,}11\ \text{ns} \\ RJ_{TX}(RMS) &= 3{,}765 \times 0{,}03/14 &&= 0{,}0081\ \text{ns} \end{aligned} \qquad (I.4)$$

For the receiver the recommendations are:

$$\begin{aligned} DJ_{RX} &= 3{,}765 \times 0{,}09 &&= 0{,}34\ \text{ns} \\ RJ_{RX}(PP) &= 3{,}765 \times 0{,}41 &&= 1{,}54\ \text{ns} \\ RJ_{RX}(RMS) &= 3{,}765 \times 0{,}41/14 &&= 0{,}110\ \text{ns} \end{aligned} \qquad (I.5)$$

Column b defines the recommendation for the serial data supplied by the chip set.

$$\begin{aligned} DJ_{SD} &= 3{,}765 \times 0{,}08 &&= 0{,}30\ \text{ns} \\ RJ_{SD}(PP) &= 3{,}765 \times 0{,}08 &&= 0{,}30\ \text{ns} \\ RJ_{SD}(RMS) &= 3{,}765 \times 0{,}08/14 &&= 0{,}022\ \text{ns} \end{aligned} \qquad (I.6)$$

The clock recovery and retiming functions of the chip set must be able to tolerate a signal which has been degraded to the level found in column c of table J.1. This is a total eye closure of 70% of a unit interval with the following components:

$$\begin{aligned} DJ_{RT} &= 3{,}765 \times 0{,}28 = 1.05\ \text{ns} \\ RJ_{RT}(PP) &= 3{,}765 \times 0{,}42 = 1.58\ \text{ns} \\ J_{RT}(TOT) &= 3{,}765 \times 0{,}70 = 2.64\ \text{ns} \end{aligned} \qquad (I.7)$$

Annex K
(informative)
FC-1 service interface

FC-1 presents a decoded-Transmission-Word-oriented service interface to FC-2. This interface uses the same primitive types as those defined by the OSI model.

The FC-1 service interface provides a conceptual view of FC-1 function from the perspective of the FC-2 level and does not restrict FCS implementation flexibility. Figure Figure K.1graphically portrays the Fibre Channel conceptual model as applied to the FC-1 service interface.

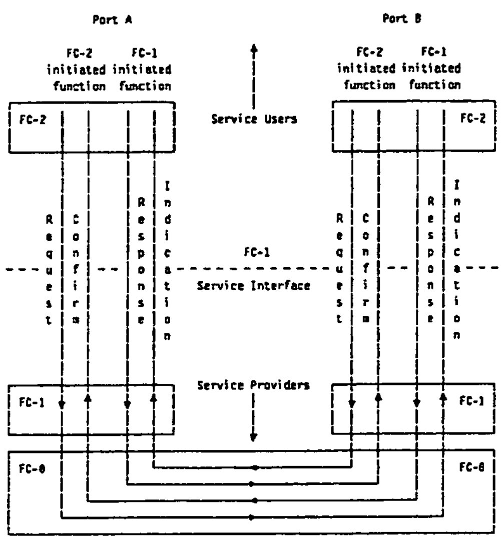


Figure K.1 - FC-1 service interface

Primitives are of four generic types:

Request The request primitive is passed from the FC-2 level to the FC-1 level to request that a service be initiated.

Indication The indication primitive is passed from the FC-1 level to the FC-2 level to indicate an internal FC-1 event which is significant to the FC-2 level. This event may be logically related to a remote service request, or may be caused by an event internal to the FC-1 level.

Response The response primitive is passed from the FC-2 level to the FC-1 level to complete a procedure previously invoked by an indication primitive.

NOTE - The response primitive is not used by the FC-1 service interface.

Confirm The confirm primitive is passed from the FC-1 level to the FC-2 level to convey the results of one or more associated previous service request(s).

Figure K.2 illustrates the primitive combinations that are supported by the FC-1 service interface. Each primitive described in this annex is correlated to one of the subfigures (a), (b), (c), or (d) of figure K.2. This figure also indicates the logical relationship of the primitive types. Primitive types that occur earlier in time and are connected by dotted lines in the diagrams are the logical antecedents of subsequent primitive types.

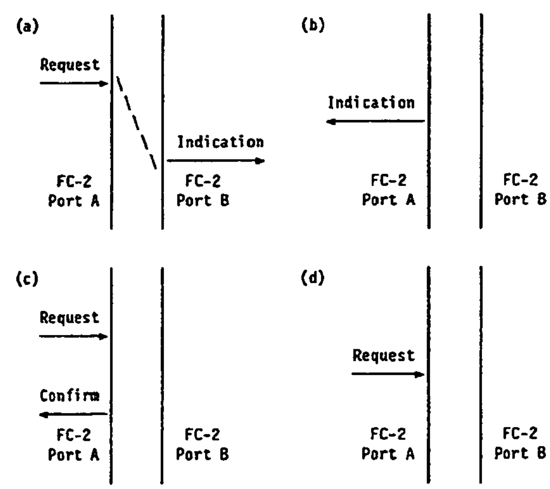


Figure K.2 - FC-1 service interface primitive combinations

The following service primitives are defined by the FC-1 service interface:

- Receiver_State
  - Indication
- FC-1_Transmitter_State

   - Indication
- FC-1_Data_Path_Width *
   - Request
   - Confirm
- FC-1_Transmitter_Clock
   - Indication
- FC-1_Data
   - Request
   - Indication
- FC-1_Receiver_Control
   - Request
- FC-1_Transmitter_Control
   - Request
- FC-1_Loopback_Control
   - Request
   - Confirm

* optional primitive

K.1 FC-1_Receiver_State.Indication

K.1.1 Function

This primitive is used by the FC-1 level to indicate the state of its receiver to an associated FC-2 level.

K.1.2 Semantics

FC-1_Receiver_State.Indication(State)

State The functional state of the FC-1 level's receiver. The allowable values of this parameter are Loss_Of_Synchronization, Synchronization_Acquired, and Reset. The format of this parameter is not defined.

K.1.3 When generated

An FC-1 level issues the FC-1_Receiver _State.Indication primitive with State = Loss_Of_Synchronization to indicate to the FC-2 level that its receiver has lost Synchronization on the signal received from its attach d Fibre (see 12.1.2.2 for a d scription of Synchronization). This state may be entered from the Synchronization-Acquired or Reset states. While in the Loss-Of-Synchronization state, the FC-1 receiver is considered Operational but does not provide FC-1_Data.Indication primitives to the FC-2 level.

An FC-1 level issues the FC-1_Receiver _State.Indication primitive with State = Synchronization_Acquired to indicate to the FC-2 level that its receiver has achieved Synchronization on the signal received from its attached Fibre (see 12.1.2.2 for a description of Synchronization). This state may be entered from the Loss-Of-Synchronization state. While in the Synchronization-Acquired state, the FC-1 receiver is considered Operational and provides FC-1_Data.Indication primitives to the FC-2 level according to the rules described in K.6.

An FC-1 level issues the FC-1_Receiver _State.Indication primitive with State = Reset to indicate to the FC-2 level that it has initiated a reset condition at its receiver. This state may be entered from any other state. While in the Reset state, the FC-1 receiver is considered Not Operational and does not provide FC-1_Data.Indication primitives to the FC-2 level.

K.1.4 Effect on receipt

Receipt of this primitive with State = Synchronization_Acquired causes the FC-2 level to prepare to accept FC-1_Data.Indication primitives from the FC-1 level. Receipt of this primitive with State = Loss_Of_Synchronization notifies the FC-2 level that FC-1_Data.Indication primitives will not be forthcoming from the FC-1 level. Receipt of this primitive with State = Reset notifies the FC-2 level that the FC-1 level is incapable of reception because it has had a receiver reset condition imposed upon it, either internally or externally.

K.1.5 Additional comments

The Loss-Of-Synchronization state is the default state of an FC-1 level's receiver upon completion of Initialization.

This primitive is illustrated by figure K.2 (b).

K.2 FC-1_Transmitter_State .Indication

K.2.1 Function

This primitive is used by the FC-1 level to indicate the state of its transmitter to an associated FC-2 level.

K.2.2 Semantics

FC-1_Transmitter_State.Indication(State)

State The functional state of the FC-1 level's transmitter. The allowable values of this parameter are Failure, Not_Enabled, Open_Fibre, and Working. The format of this parameter is not defined.

K.2.3 When generated

An FC-1 level issues the FC-1_Transmitter _State.Indication primitive with State = Failure to indicate to the FC-2 level that its transmitter has entered the Failure state as described by 12.3.5. This state may be entered from the Working state. While in the Failure state, the FC-1 transmitter is considered Not Operational, does not accept FC-1_Data.Request primitives from the FC-2 level, and does not provide FC-1_Transmitter _Clock.Indication primitives to the FC-2 level.

An FC-1 level issues the FC-1_Transmitter _State.Indication primitive with State = Not_Enabled to indicate to the FC-2 level that its transmitter is Operational but is prevented from transmitting a signal onto its associated Fibre. This state may be entered from the Working or Open-Fibre states. It may result from the processing of an FC-1_Transmitter_Control.Request primitive or from an external event (e.g., FC-2 link reset, removal of laser safety condition when a transmitter disable request remains outstanding). While in the Not-Enabled state, the FC-1 transmitter is considered Operational and accepts FC-1_Data.Request primitives from the FC-2 level in synchronization with the FC-1_Transmitter_Clock.Indication primitives provided by it to the FC-2 level. However, no signals resulting from the accepted FC-1_Data.Request primitives are transmitted onto its associated Fibre. If the FC-1 level is in Loopback mode, the encoded bit stream resulting from the accepted FC-1_Data.Request primitives is provided to the FC-1 receiver.

An FC-1 level issues the FC-1_Transmitter _State.Indication primitive with State = Open_Fibre to indicate to the FC-2 level that its transmitter is Operational but is prevented from transmitting a signal onto its associated Fibre. This state may be entered from the Working or Not-Enabled states. It results from detection by FC-0 laser safety procedures of a laser safety condition (i.e., a condition which requires that the transmitter cease transmission). While in the Open-Fibre state, the FC-1 transmitter is considered Operational and accepts FC-1_Data.Request primitives from the FC-2 level in synchronization with the FC-1_Transmitter _Clock.Indication primitives provided by it to the FC-2 level. However, no signals resulting from the accepted FC-1_Data.Request primitives are transmitted onto its associated Fibre. If the FC-1 level is in Loopback mode, the encoded bit stream resulting from the accepted FC-1_Data.Request primitives is provided to the FC-1 receiver.

An FC-1 level issues the FC-1_Transmitter _State.Indication primitive with State = Working to indicate to the FC-2 level that its transmitter is operating normally according to the error-monitoring procedure described by 12.3.5. This state may be entered from the Not-Enabled or Open-Fibre states. While in the Working state, the FC-1 transmitter is considered Operational and accepts FC-1_Data.Request primitives from the FC-2 level in synchronization with the FC-1_Transmitter_Clock.Indication primitives provided by it to the FC-2 level. Signals resulting from the accepted FC-1_Data.Request primitives are transmitted onto its associated Fibre. If the FC-1 level is in Loopback mode, the encoded bit stream resulting from the accepted FC-1_Data.Request primitives are also provided to the FC-1 receiver.

K.2.4 Effect on receipt

Receipt of this primitive with State = Working causes the FC-2 level to continue issuance of FC-1_Data.Request primitives to the FC-1 level (synchronous to FC-1_Transmitter_Clock .Indication primitives from the FC-1 level). Receipt of this primitive with State = Not_Enabled causes the FC-2 level to begin or continu issuance of FC-1_Data.Request primitives to the FC-1 level

(synchronous to FC-1_Transmitter_Clock .Indication primitives from the FC-1 level). No transmitted signals result from the receipt of FC-1_Data.Request primitives. The transmitter remains Operational. Receipt of this primitive with State = Open_Fibre causes the FC-2 level to begin or continue issuance of FC-1_Data.Request primitives to the FC-1 level (synchronous to FC-1_Transmitter_Clock .Indication primitives from the FC-1 level). No transmitted signals result from the receipt of FC-1_Data.Request primitives. The transmitter remains Operational. Receipt of this primitive with State = Failure notifies the FC-2 level that FC-1_Transmitter_Clock.Indication primitives will not be forthcoming from the FC-1 level and that FC-1_Data.Request primitives will not be accepted by the FC-1 level. The FC-2 level may initiate maintenance activity.

K.2.5 Additional comments

The Not-Enabled state is the default state of an FC-1 level's transmitter upon completion of Initialization.

This primitive is illustrated by figure K.2 (b).

K.3 FC-1_Data_Path_Width.Request

K.3.1 Function

This primitive is used to establish the path width of a variable-width data interface between an FC-1 and an FC-2 level.

K.3.2 Semantics

FC-1_Data_Path_Width.Request(Width)

Width The data path width of the interface between the FC-1 and FC-2 levels. The allowable values of this parameter are Byte, Half_Word, and Word. The format of this parameter is not defined.

K.3.3 When generated

An FC-2 level issues the FC-1_Data_Path_Width.Request primitive to specify to the FC-1 level the width of the data path to be used (Byte, Half_Word, or Word) between the FC-1 and FC-2 levels.

K.3.4 Eff ct on r ceipt

Receipt of the FC-1_Data_Path_Width.Request primitive causes an FC-1 level to attempt to establish the requested path width for its data interface to FC-2.

K.3.5 Additional comments

This primitive is not required for those implementations which use an FC-1 level with a fixed data path width. For those implementations which use an FC-1 level with a variable data path width, the attached FC-2 level issues this primitive to indicate the preferred path width.

An FC-1_Data_Path_Width.Request primitive is accepted by an FC-1 level or a fixed data path width is defined before FC-1_Data and FC-1_Transmitter_Clock primitives can be issued.

The Half_Word value of Width corresponds to a 2-byte interface. The Word value of Width corresponds to a 4-byte interface.

This primitive is illustrated by the request portion of figure K.2 (c).

K.4 FC-1_Data_Path_Width.Confirm

K.4.1 Function

This primitive is used to accept or reject the establishment of the path width of a variable-width data interface between an FC-1 and an FC-2 level.

K.4.2 Semantics

FC-1_Data_Path_Width.Confirm(Status)

Status The results of the FC-1_Data_Path_Width.Request primitive. The allowable values of this parameter are Accept and Reject. The format of this parameter is not defined.

K.4.3 When generated

An FC-1 level issues the FC-1_Data_Path_Width.Confirm primitive to an FC-2 level to indicate the acceptance or rejection of a previously issued FC-1_Data_Path_Width.Request primitive.

K.4.4 Eff ct on r ceipt

Receipt of the FC-1_Data_Path_Width.Indication primitive with Status = Accept informs the FC-2 level that path width establishment was successful. Receipt of the FC-1_Data_Path_Width.Indication primitive with Status = Reject informs the FC-2 level that path width establishment was unsuccessful.

K.4.5 Additional comments

This primitive is not required for those implementations which use an FC-1 level with a fixed data path width. For those implementations which use an FC-1 level with a variable data path width, the attached FC-1 level must issue this primitive in response to an FC-1_Data_Path_Width.Request primitive.

This primitive is illustrated by the confirm portion of figure K.2 (c).

K.5 FC-1_Transmitter_Clock .Indication

K.5.1 Function

This primitive is used by the FC-1 level to provide data transmission synchronization information, including an indication of Transmission Word alignment boundaries, to the FC-2 level.

K.5.2 Semantics

FC-1_Transmitter _Clock.Indication(Boundary)

Boundary An indication of the transmission boundary presented by the transmitter clock. The allowable values of this parameter are Word and Null. The format of this parameter is not defined.

K.5.3 When generated

An FC-1 level with a transmitter in the Working, Open-Fibre, or Not-Enabled states issues the FC-1_Transmitter_Clock.Indication primitive to provide a data transmission synchronization signal to the FC-2 level. Synchronous to the receipt of an FC-1_Transmitter_Clock.Indication primitive, the FC-2 level may choose to issue an FC-1_Data.Request primitive to the FC-1 level to present data to be transmitted (see K.6 for a description of the rules associated with data transmission).

The FC-1_Transmitter_Clock.Indication primitive is issued periodically to the FC-2 level with a frequency based upon the FC-0 transmission frequency and the data path width between the FC-1 and FC-2 levels. When a byte-wide data path is provided, the frequency of issuance = FC-0 transmission frequency / 10. When a half-word-wide data path is provided, the frequency of issuance = FC-0 transmission frequency / 20. When a word-wide data path is provided, the frequency of issuance = FC-0 transmission frequency / 40.

The Boundary parameter is used by the FC-1 level to indicate whether the transmission data presented on an FC-1_Data.Request primitive that is associated with the FC-1_Transmitter_Clock.Indication primitive may be of the type that requires a Transmission-Word boundary. When Boundary = Word, any information type (Valid_Data_Byte, Delimiter, or Primitive_Signal) may be presented on the Type1 parameter of the associated FC-1_Data.Request primitive. When Boundary = Null, only the type Valid_Data_Byte may be presented on the Type1 parameter of the associated FC-1_Data.Request primitive.

Transmission-Word alignment is established by an FC-1 transmitter at the time it enters the working state and begins to transmit idle words as specified in K.6. The established Transmission-Word alignment is indicated by the FC-1 level to the FC-2 level through the FC-1_Transmitter_Clock.Indication primitive. When a byte-wide data path width has been specified, the FC-1 level indicates a Transmission-Word boundary on every fourth FC-1_Transmitter_Clock.Indication primitive. When a half-word-wide data path width has been specified, the FC-1 level indicates a Transmission-Word boundary on every second FC-1_Transmitter_Clock.Indication primitive. When a word-wide data path width has been specified, the FC-1 level indicates a Transmission- Word boundary on every FC-1_Transmitter_Clock.Indication primitive.

K.5.4 Effect on receipt

Receipt of this primitive informs an FC-2 level that a request for information transmission can occur via issuance of an FC-1_Data.Request primitive.

K.5.5 Additional comments

This primitive is illustrated by figure K.2 (b).

K.6 FC-1_Data.Request

K.6.1 Function

This primitive is used by the FC-2 level to pass transmission requests to an associated FC-1 level.

K.6.2 Semantics

```
FC-1_Data.Request(Type1, Code1
                  Type2, Code2
                  Type3, Code3
                  Type4, Code4)
```

Type The type of transmission information that is to be encoded and transmitted by the FC-1 level. The allowable values of this parameter are Valid_Data_Byte, Non_-Repeating_Ordered_Set, and Repeating_-Ordered_Set. The format of this parameter is not defined.

When a Non-Repeating Ordered Set or Repeating Ordered Set is indicated, a single Type field (Type1) is provided regardless of data path width. When a Valid Data Byte is indicated, a separate Type field is provided for each byte of the data path (i.e., Type1 is provided when a byte-wide data path width is specified, Type1 and Type2 are provided when a half-word-wide data path width is specified, and Type1, Type2, Type3, and Type4 are provided when a word-wide data path width is specified).

The values Non_Repeating_Ordered_Set and Rep ating_Ordered_Set are restricted to the Type1 parameter. When present, the Type2, Type3, and Type4 parameters are set to the Valid_Data_-Byte value.

When multiple Type parameters are presented in an FC-1_Data.Request primitive, the corresponding Transmission Characters are transmitted in the order given (i.e., Type1 corresponds to the first character received, followed by Type2, Type3 (if present), and Type4 (if present)).

Code The Valid Data Byte or Ordered Set that is to be encoded and transmitted by the FC-1 level. The contents of the Code parameter are defined as follows:

– When Type = Valid_Data_Byte, the Code parameter contains information in a format not defined. Standard which indicates the Valid Data Byte.

– When Type = Non_Repeating_Ordered_Set the Code parameter contains information in a format not defined. which indicates the Non-Repeating Ordered Set. All Ordered Sets specified in 11.4, "Ordered Sets" may be specified as a Non-Repeating Ordered Set.

– When Type = Repeating_Ordered_Set, the Code parameter contains information in a format not defined. Repeating Ordered Set. FC-2 must guarantee that only Primitive Sequences and the Idle Primitive Signal are specified as Repeating Ordered Sets.

When a Repeating Ordered Set or Non-Repeating Ordered Set is indicated, a single Code field (Code1) is provided regardless of data path width. When a Valid Data Byte is indicated by Type, a separate Code field is provided for each byte of the data path (i.e., Code1 is provided when a byte-wide data path width is specified, Code1 and Code2 are provided when a half-word-wide data path width is specified, and Code1, Code2, Code3, and Code4 are provided when a word-wide data path width is specified).

When multiple Code parameters are presented in an FC-1_Data.Request primitive, the corresponding Transmission Charact rs are transmitted in the order given (i. ., Code1 corresponds to th first character received, followed by Code2, Code3 (if present), and Code4 (if present)).

K.6.3 When generated

An FC-2 level issues the FC-1_Data.Request primitive to the FC-1 level synchronous to a received FC-1_Transmitter_Clock.Indication primitive when it wishes to specify information to be encoded and transmitted by the FC-1 level. Information to be encoded is passed by the FC-2 level to the FC-1 level according to the following rules:

a) When the data path width between the FC-1 and FC-2 levels is defined to be Byte, the information passed by the FC-1_Data.Request primitive may be a Valid Data Byte, a Non-Repeating Ordered Set, or a Repeating Ordered Set. If a Repeating Ordered Set or Non-Repeating Ordered Set is passed by the FC-1_Data.Request primitive, the FC-2 level does not issue an FC-1_Data.Request primitive synchronous to the following three received FC-1_Transmitter_Clock.Indication primitives.

b) When the data path width between the FC-1 and FC-2 levels is defined to be Half_Word, the information passed by the FC-1_Data.Request primitive may be two Valid Data Bytes, a Non-Repeating Ordered Set, or a Repeating Ordered Set. If a Repeating Ordered Set or Non-Repeating Ordered Set is passed by the FC-1_Data.Request primitive, the FC-2 level does not issue an FC-1_Data.Request primitive synchronous to the following received FC-1_Transmitter _Clock.Indication primitive.

c) When the data path width between the FC-1 and FC-2 levels is defined to be Word, the information passed by the FC-1_Data.Request primitive may be four Valid Data Bytes, a Repeating Ordered Set, or a Non-Repeating Ordered Set.

The Boundary parameter of the FC-1_Transmitter _Clock.Indication primitive may restrict what an FC-2 level is allowed to present on an FC-1_Data.Request primitive. When Boundary = Word, any information type (Valid_Data_Byte, Non_Repeating_ Ordered_Set, or Repeating_ Ordered_Set may be presented on the Type1 parameter of the associated FC-1_Data.Request primitive. When Boundary = Null, only Valid_-Data_Byte may be presented on the Type1 parameter of the associated FC-1_Data.Request primitive. When information types other than Valid_Data_Byte are presented, unpredictable transmitter behavior results and the transmitter enters the Failure state.

K.6.4 Effect on receipt

Receipt of an FC-1_Data.Request primitive by an FC-1 transmitter in the Not-Enabled or Open-Fibre state causes the FC-1 level to attempt to encode the data indicated by the primitive. If the FC-1 level is in Loopback mode, data are provided to the FC-1 receiver for decoding and presentation to the FC-2 level. No signal is transmitted onto the attached Fibre as the result of a FC-1_Data.Request primitive received by an FC-1 transmitter in the Not-Enabled state.

Receipt of an FC-1_Data.Request primitive by an FC-1 transmitter in the Working state causes the FC-1 level to attempt to encode and transmit the data indicated by the primitive onto its attached Fibre. If the FC-1 level is in Loopback mode, this data is also provided to the FC-1 receiver for decoding and presentation to the FC-2 level.

Encoded information is processed by an FC-1 transmitter in the Working, Not-Enabled, or Open-Fibre state according to the following rules:

a) Upon entering the Working state, the FC-1 transmitter must be presented with a proper FC-1_Data.Request primitive by FC-2.

b) Upon receipt of an FC-1_Data.Request primitive specifying Repeating_Ordered_Set, the FC-1 transmitter begins transmitting and/or providing the specified encoded Ordered Set and continues to transmit and/or provides this Ordered Set until receipt of a subsequent FC-1_Data.Request primitive.

c) Upon receipt of an FC-1_Data.Request primitive specifying Valid_Data_Byte or Non_-Repeating_Ordered_Set the FC-1 transmitter encodes and transmits and/or provides the specified information. Receipt of such an FC-1_Data.Request primitive ends the continuous transmission and/or provision of Repeating Ordered Sets if such transmission and/or provision has been previously established by a prior FC-1_Data.Request primitive.

Receipt of an FC-1_Data.R quest primitive not conforming to the rules described previously results in unpredictable FC-1 level behavior.

Encoded information is not transmitt d and/or provided and FC-1_Data.Request primitives are not accepted by an FC-1 transmitter in the Failure state.

K.6.5 Additional comments

This primitive is illustrated by the request portion of figure K.2 (a).

K.7 FC-1_Data.Indication

K.7.1 Function

This primitive is used by the FC-1 level to pass received and decoded data to an associated FC-2 level.

K.7.2 Semantics

FC-1_Data.Indication(Type1, Code1
Type2, Code2
Type3, Code3
Type4, Code4)

Type The type of transmission information that has been received and decoded by the FC-1 level. The allowable values of this parameter are

Valid_Data_Byte,
Ordered_Set,
Special_Code,
Code_Violation,
Invalid_Special_Code_Alignment, and
Invalid_Beginning_Running_Disparity.

The latter four values are associated with the detection of invalid Transmission Words as specified by "Invalid Transmission Word rules." Special Codes and Valid Data Bytes may be reported in conjunction with these values as defined by the rules specified in K.7.3. The format of this parameter is not defined.

When an Ordered Set is indicated, a single Type field (Type1) is provided regardless of data path width. When information other than an Ordered Set is indicated, a separate Type field is provided for each byte of the data path (i.e., Type1 is provided when a byte-wide data path width is specified, Type1 and Type2 are provided when a half-word-wide data path width is specified, and Type1, Type2, Type3, and Type4 are provided when a word-wide data path width is specified).

The values Ordered Set and Invalid_ Beginning_Running_Disparity are restricted to the Type1 parameter. When present, the Type2, Type3, and Type4 parameters are set to the Valid_Data_-Byte, Code_Violation, Special_Code, or Invalid_ Special_Code_Alignment value.

When multiple Type parameters are presented in an FC-1_Data.Indication primitive, the corresponding Transmission Characters were received in the order given (i.e., Type1 corresponds to the first character received, followed by Type2, Type3 (if present), and Type4 (if present)).

Code The information (Valid Data Byte, Ordered Set, or invalid information) that has been received and decoded by the FC-1 level. The contents of the Code parameter are defined as follows:

— When Type = Valid_Data_Byte, the Code parameter contains information in a format not defined. which indicates the Valid Data Byte.

— When Type = Special_Code, the Code parameter contains information in a format not defined. which indicates the Special Code.

— When Type = Ordered_Set, the Code parameter contains information in a format not defined. which indicates the Ordered Set.

— When Type = Invalid_ Special_Code_Alignment, the Code parameter contains information in a format not defined. which indicates the Special Code that was detected in an invalid position in the second, third, or fourth character of the Transmission Word.

— When Type = Code_Violation, the Code parameter is not meaningful and is ignored.

— When Type = Invalid_ Beginning_Running_Disparity, the Code parameter contains information in a format not defined. which indi-

cates the otherwise valid Ordered Set which was rec ived with improper Beginning Running Disparity.

When an Ordered Set or an indication of invalid Beginning Running Disparity is indicated, a single Code field (Code1) is provided regardless of data path width. When information other than an Ordered Set or an indication of invalid Beginning Running Disparity is indicated by Type, a separate Code field is provided for each byte of the data path (i.e., Code1 is provided when a byte-wide data path width is specified, Code1 and Code2 are provided when a half-word-wide data path width is specified, and Code1, Code2, Code3, and Code4 are provided when a word-wide data path width is specified). Note that the Code field associated with a Type field indicating Code_Violation, while present, is not meaningful.

When multiple Code parameters are presented in an FC-1_Data.Indication primitive, the corresponding Transmission Characters were received in the order given (i.e., Code1 corresponds to the first character received, followed by Code2, Code3 (if present), and Code4 (if present)).

K.7.3 When generated

An FC-1 receiver in the Synchronization-Acquired state issues the FC-1_Data.Indication primitive to pass received and decoded Transmission Words to the FC-2 level. When the FC-1 level is in normal mode, Transmission Words are received from the Fibre attached to the FC-1 receiver. When the FC-1 level is in Loopback mode, received Transmission Words are provided by the FC-1 transmitter.

Decoded Transmission Words that are determined to be valid by the FC-1 level are passed to the FC-2 level according to the following rules:

a) When the data path width between the FC-1 and FC-2 levels is defined to be Byte, the information passed by the FC-1_Data.Indication primitive may be a Valid Data Byte, a Special Code, or an Ordered Set (a Special Code is passed only when an Unrecognized Ordered Set is received and decoded by the receiver).

b) When the data path width between the FC-1 and FC-2 levels is defined to be Half_Word, the information passed by the FC-1_Data.Indication primitive may be two Valid Data Bytes, a Special Code followed by a Valid Data Byte, or an Ordered Set (the Special Code / Valid Data Byte pair is passed only when an Unrecognized Ordered Set is received and decoded by the receiver).

c) When the data path width between the FC-1 and FC-2 levels is defined to be Word, the information passed by the FC-1_Data.Indication primitive may be four Valid Data Bytes, a Special Code followed by three Valid Data Bytes, or an Ordered Set (the Special Code / Valid Data Byte / Valid Data Byte / Valid Data Byte string is passed only when an Unrecognized Ordered Set is received and decoded by the receiver).

Decoded Transmission Words that are determined to be invalid by the FC-1 level are passed to the FC-2 level according to the following rules:

a) When the data path width between the FC-1 and FC-2 levels is defined to be Byte, the information passed by the FC-1_Data.Indication primitive may be one of the following: a Code Violation, an invalid Special Code alignment, or an indication of invalid Beginning Running Disparity. Preceding or subsequent FC-1_Data.Indication primitives associated with the invalid Transmission Word may contain Valid Data Bytes and/or Special Codes (note that an invalid Transmission Word as specified by "Invalid Transmission Word rules" may contain one or more valid Transmission Characters; it is these characters that are passed as Valid Data Bytes or Special Codes in the four FC-1_Data.Indication primitives which represent the invalid Transmission Word).

b) When the data path width between the FC-1 and FC-2 levels is defined to be Half_Word, the information passed by the FC-1_Data.Indication primitive may be one or more of the following: a Code Violation, an invalid Special Code alignment, or an indication of invalid Beginning Running Disparity. Indications of invalid Beginning Running Disparity may b specified only in the Type1 parameter. Code Violations and invalid Special Code alignments may be specified in either or both of the Type1 and Type2 parameters. These values may be combined with a

Valid Data Byte or Special Code value in one of the Type1 and Type2 parameters according to the contents of the received and decoded Transmission Word and the usage rules described in the definition of the Type parameter. Preceding or subsequent FC-1_Data.Indication primitives associated with the invalid Transmission Word may also contain Valid Data Bytes and/or Special Codes (note that an invalid Transmission Word as specified by "Invalid Transmission Word rules" may contain one or more valid Transmission Characters; it is these characters that are passed as Valid Data Bytes or Special Codes in the two FC-1_Data.Indication primitives which represent the invalid Transmission Word).

c) When the data path width between the FC-1 and FC-2 levels is defined to be Word, the information passed by the FC-1_Data.Indication primitive may be one or more of the following: a Code Violation, an invalid Special Code alignment, or an indication of invalid Beginning Running Disparity. Indications of invalid Beginning Running Disparity may be specified only in the Type1 parameter. Code Violations may be specified in one or more of the Type1, Type2, Type3, and Type4 parameters. Invalid Special Code alignments may be specified in one or more of the Type2, Type3, and Type4 parameters. These values may be combined with Valid Data Byte and Special Code values in one or more of the Type1, Type2, Type3, and Type4 parameters according to the contents of the received and decoded Transmission Word and the usage rules described in the definition of the Type parameter (note that an invalid Transmission Word as specified by "Invalid Transmission Word rules" may contain one or more valid Transmission Characters; it is these characters that are passed as Valid Data Bytes or Special Codes in the FC-1_Data.Indication primitive which represents the invalid Transmission Word).

When information other than Ordered Sets or an indication of invalid Beginning Running Disparity is received and decoded by the FC-1 level, the FC-1_Data.Indication primitive is issued periodically to the FC-2 l vel with a frequency based upon the FC-0 transmission frequency and the data path width between the FC-1 and FC-2 levels. When a byte-wide data path is provided, the frequency of issuance = FC-0 transmission frequency / 10. When a half-word-wide data path is provided, the frequency of issuance = FC-0 transmission frequency / 20. When a word-wide data path is provided, the frequency of issuance = FC-0 transmission frequency / 40. When an Ordered Set or an indication of invalid Beginning Running Disparity is received by the FC-1 level, the FC-1_Data.Indication primitive is issued according to the rules described below.

Received and decoded information is passed to the FC-2 level by an FC-1 receiver in the Synchronization-Acquired state according to the following rules:

a) Upon entering the Synchronization-Acquired state, the FC-1 receiver indicates receipt of the Ordered Set which allowed it to enter the Synchronization-Acquired state. When the Ordered Set is recognized, this indication is completed by issuing an FC-1_Data.Indication with Type = Ordered_Set and Code indicating the appropriate value. When the Ordered Set is unrecognized, this indication is completed by issuing one or more FC-1_Data.Indication primitives sufficient to represent the Unrecognized Ordered Set according to the defined data path width between the FC-1 and FC-2 levels.

b) Upon receipt of an incoming bit stream representing a recognized Ordered Set, the FC-1 receiver indicates receipt of this Ordered Set by issuing an FC-1_Data_Indication with Type = Ordered_Set and Code set to the appropriate value.

c) Upon receipt of an incoming bit stream representing an Unrecognized Ordered Set, the FC-1 receiver indicates receipt of this Ordered Set by issuing one or more FC-1_Data.Indication primitives sufficient to represent the Unrecognized Ordered Set according to the defined data path width between the FC-1 and FC-2 levels.

d) Upon receipt of an incoming bit stream representing a data character that is not part of an Ordered Set, the FC-1 receiver indicates receipt of this character by issuing an FC-1_Data.Indication with Type = Valid_-Data_Byte and Code set to the appropriate value. Additional Valid_Data_Byte ntries are provided when the data path width between the FC-1 and FC-2 levels is Half_Word or Word.

e) Upon receipt of an incoming bit stream representing an invalid Transmission Word as specified by "Invalid Transmission Word rules," the FC-1 receiver indicates receipt of this word by issuing an FC-1_Data.Indication with Type indicating the appropriate condition or conditions according to the rules described previously in this annex.

Decoded information is not passed to the FC-2 level and FC-1_Data.Indication primitives are not issued by an FC-1 receiver in the Loss-Of-Synchronization or Reset states.

K.7.4 Effect on receipt

Receipt of an FC-1_Data.Indication primitive by an FC-2 level causes the FC-2 level to accept the received and decoded data indicated by the primitive from the FC-1 level.

K.7.5 Additional comments

This primitive is illustrated by the indication portion of figure K.2 (a).

K.8 FC-1_Receiver_Control.Request

K.8.1 Function

This primitive is used by the FC-2 level to request that the FC-1 receiver be reset.

K.8.2 Semantics

FC-1_Receiver_Control.Request(Control)

Control The receiver control request made by the FC-2 level. The allowable value of this parameter is Reset. The format of this parameter is not defined.

K.8.3 When generated

An FC-2 level issues the FC-1_Receiver_Control.Request primitive with Control = Reset to request that the FC-1 level receiver be reset.

K.8.4 Effect n receipt

Receipt of this primitive with Control = Reset causes the FC-1 level to initiate a reset condition in its receiver. If the receiver is not already in the reset state, the FC-1 level concurrently issues an FC-1_Receiver _State.Indication primitive to the FC-2 level with State = Reset. The FC-1 level is incapable of providing FC-1_Data.Indication primitives to the FC-2 level while a reset condition exists in its receiver. When the receiver reset condition is exited, the FC-1 level issues an FC-1_Receiver _State .Indication primitive to the FC-2 level with State = Loss_Of_Synchronization.

K.8.5 Additional comments

An FC-1 reset condition may result in a request to the FC-0 receiver to attempt to reacquire bit synchronization.

This primitive is illustrated by figure K.2 (d).

K.9 FC-1_Transmitter_Control .Request

K.9.1 Function

This primitive is used by the FC-2 level to request that the FC-1 transmitter be enabled or disabled.

K.9.2 Semantics

FC-1_Transmitter _Control.Request(Control)

Control The enable control request made by the FC-2 level. The allowable values of this parameter are Enable and Disable. The format of this parameter is not defined.

K.9.3 When generated

An FC-2 level issues the FC-1_Transmitter _Control.Request primitive with Control = Enable to request that the FC-1 level transmitter be enabled.

An FC-2 level issues the FC-1_Transmitter _Control.Request primitive with Control = Disable to request that the FC-1 level transmitter be disabled.

K.9.4 Effect n rec ipt

Receipt of this primitive with Control = Enable causes the FC-1 level to enable its transmitter unless the transmitter is in one of the following conditions:

- The transmitter is in the Failure state.
- The transmitter has been placed in the Open-Fibre state as a result of detection by FC-0 laser safety procedures of a laser safety condition (i.e., a condition which requires that the transmitter cease transmission).

If the transmitter is not already enabled and it is not in one of the above conditions, the FC-1 level concurrently issues an FC-1_Transmitter_State Indication primitive to the FC-2 level with State = Working. Otherwise, receipt of this primitive with Control = Enable by a transmitter does not result in a transmitter state change. However, if a previously-detected laser safety condition is present at the time of receipt, the state transition which results when the laser safety condition is removed may be affected (see 12.4).

Receipt of this primitive with Control = Disable causes the FC-1 level to disable its transmitter if not previously disabled. If the transmitter is in the Working state at the time of receipt, the FC-1 level concurrently issues an FC-1_Transmitter_State.Indication primitive to the FC-2 level with State = Not_Enabled. Otherwise, receipt of this primitive with Control = Disable by a transmitter does not result in a transmitter state change. However, if a previously-detected laser safety condition is present at the time of receipt (i.e., the transmitter is in the Open-Fibre state), the state transition which results when the laser safety condition is removed may be affected (see 12 4).

K.9.5 Additional comments

This primitive is illustrated by figure K.2 (d).

K.10 FC-1_L pback_Control .Request

K.10.1 Function

This primitive is used by the FC-2 level to request that the FC-1 Loopback function be enabled or disabled.

K.10.2 Semantics

FC-1_Loopback_Control.Request(Control)

Control The Loopback control request made by the FC-2 level. The allowable values of this parameter are Enable and Disable. The format of this parameter is not defined.

K.10.3 When generated

An FC-2 level issues the FC-1_Loopback_ Control.Request primitive with Control = Enable to request that the FC-1 Loopback function be enabled.

An FC-2 level issues the FC-1_Loopback_ Control.Request primitive with Control = Disable to request that the FC-1 Loopback function be disabled.

K.10.4 Effect on receipt

Receipt of this primitive with Control = Enable causes the FC-1 level to enable its Loopback function unless the FC-1 transmitter or receiver is in a state not compatible with the Loopback function. When the Loopback function is enabled, information passed to the FC-1 level via FC-1_Data.Request primitives is shunted to the FC-1 receiver and reflected in FC-1_Data.Indication primitives issued by the receiver after Synchronization is reestablished (if necessary), overriding any information received by the FC-1 receiver.

Receipt of this primitive with Control = Disable causes the FC-1 level to disable its Loopback function. When the Loopback function is disabled, information passed to the FC-1 level via FC-1_Data.Request primitives is encoded and transmitted by the FC-1 transmitter. Information received and decoded by the FC-1 receiver is indicated to the FC-2 level by

FC-1_Data.Indication primitives after Synchronization is reestablished (if necessary).

K.10.5 Additional comments

The FC-1 Loopback function is disabled upon completion of Initialization of an FC-1 transmitter and receiver.

Behavior of a transmitter in the Working state is unpredictable when the FC-1 Loopback function is enabled.

Loss of Synchronization may occur whenever the Loopback function is enabled or disabled.

This primitive is illustrated by the request portion of figure K.2 (c).

K.11 FC-1_Loopback_Control.Confirm

K.11.1 Function

This primitive is used by the FC-1 level to accept or reject the request that the FC-1 Loopback function be enabled or disabled.

K.11.2 Semantics

FC-1_Loopback_Control.Confirm(Status)

Status The results of the FC-1_Loopback_ Control.Request primitive. The allowable values of this parameter are Accept and Reject. The format of this parameter is not defined.

K.11.3 When generated

An FC-1 level issues the FC-1_Loopback_ Control.Confirm primitive to an FC-2 level to indicate the acceptance or rejection of a previously issued FC-1_Loopback_ Control.Request primitive. Acceptance is indicated only when valid information is presented by the receiver via FC-1_Data.Indication primitives.

K.11.4 Effect on receipt

Receipt of the FC-1_Loopback_Control.Indication primitive with Status = Accept informs the FC-2 level that the enabling or disabling of the FC-1 Loopback function was successful or that the FC-1 Loopback function was already enabled or disabled.

Receipt of the FC-1_Loopback_Control.Indication primitive with Status = Reject informs the FC-2 level that the enabling of the FC-1 Loopback function was unsuccessful. A request to enable the FC-1 Loopback function is rejected whenever the FC-1 receiver is not in the Synchronization-Acquired or Loss-Of-Synchronization state or the FC-1 transmitter is not in the Working, Open-Fibre, or Not-Enabled state. (a request to disable the FC-1 Loopback function is always accepted).

K.11.5 Additional comments

This primitive is illustrated by the confirm portion of figure K.2 (c).

Annex L
(informative)
Communication models

L.1 Model 1 example

When one port of the channel only transmits Data frames and the other port only transmits Link level (Link_Control) responses, the channel is said to be performing half-duplex communication.

For example in figure L.1,

- N_Port A(1) transmits a Data frame (command or data) on its outbound Fibre and N_Port B(1) receives the Data frame on its inbound Fibre.
- In response, N_Port B(1) transmits a Link_Control frame concurrently on its outbound Fibre and N_Port A(1) receives it on its inbound Fibre.
- Multiple Data frames from N_Port A(1) and the respective Link_Control frames from N_Port B(1) results in similar flow.

Figure L.1 depicts the FC-2 half-duplex communication model with its logical structure and components. The model is composed of the following components:

- Node A and B
- N_Ports A(1) and B(1)
- Link_Control_Facility (LCF) in each N_Port
- Originator and Originator status in one N_Port
- Responder and Responder status in the other N_Port

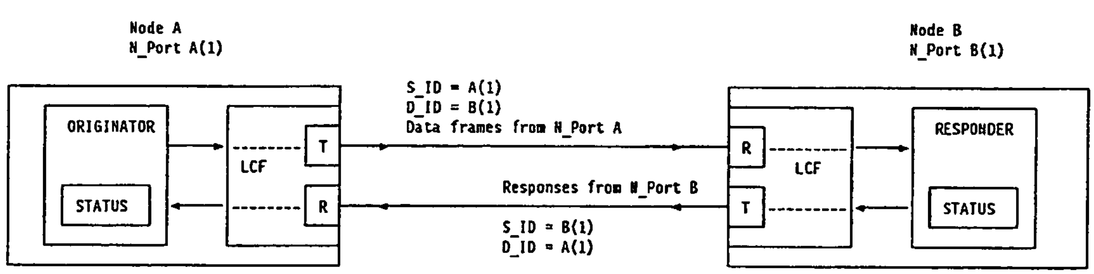
The Originator is the logical function in an N_Port A(1) which initiates an Exchange, whereas the Responder is the logical function in an N_Port which is the destination of the Exchange. Either N_Port can be an Originator and the other N_Port a Responder in figure L.1 at any given time. The role of N_Ports as Originator and Responder do not change during a given Exchange.

L.2 Model 2 example

When both ports of a channel transmit Data frames and Link_Control frames (responses to Data frames) in directions opposite to their respective Data frames, the channel is said to be performing duplex communication.

For example in figure L.2,

- N_Port A(1) transmits a Data frame on its outbound Fibre and N_Port B(1) receives it on its inbound Fibre.
- Simultaneously N_Port B(1) transmits its Data frame to N_Port A(1) on its outbound Fibre and N_Port A(1) receives it on its inbound Fibre.
- In response to the Data frame from N_Port A(1), N_Port B(1) transmits a Link_Control response on its outbound Fibre and N_Port A(1) receives the response on its inbound Fibre.
- In response to the Data frame from N_Port B(1), N_Port A(1) transmits a Link_Control response on its outbound Fibre and N_Port B(1) receives the response on its inbound Fibre.



Legend: S_ID = Source Identifier
D_ID = Destination Identifier
LCF = Link_Control_Facility

Figure L.1 - FC-2 half-duplex communicati n model example

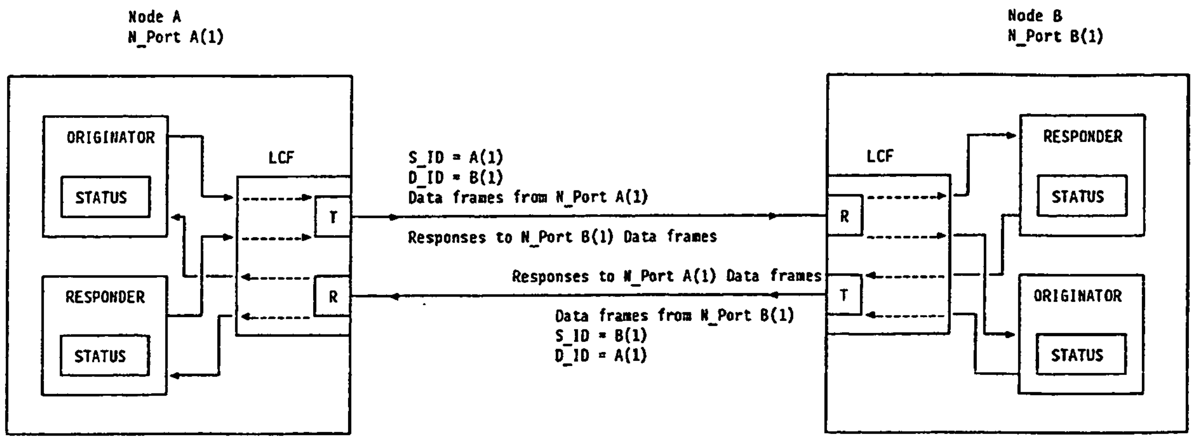


Figure L.2 - FC-2 duplex communication model example

— Multiple Data frames from N_Port A(1), and N_Port B(1) and the respective responses from N_Port B(1), and N_Port A(1) results in similar flow.

Figure L.2 depicts the FC-2 duplex communication model with its logical structure and components. The FC-2 duplex communication model is bidirectionally symmetrical and both communicating N_Ports have equivalent abilities.

The FC-2 duplex communication model consists of an Originator and a Responder in each of the communicating N_Ports.

From the perspective of a given N_Port, the Outbound Fibre transmits a mixture of Data frames according to the activity generated by the N_Port and Link level responses to the Data frames received on the Inbound Fibre from the remote N_Port.

L.3 Model 3 example

When the channel transmits Data frames in one direction with the Link_Control frames arriving in the opposite direction, and limits the flow of Data frames and Link_Control frames at any given time to a single direction, the channel is said to be performing dual-simplex communication.

For example in figure L.3,

— N_Port A(1) transmits a Data frame on its outbound Fibre and N_Port B(1) receives the Data frame on its inbound Fibre through the unidirectional Fabric. (simplex from A(1) to B(1)).
— In response to the Data frame, N_Port B(1) transmits a Link_Control response on its outbound Fibre and, due to unidirectional limitation of the Fabric, N_Port A(1) receives the response non-concurrently on its inbound Fibre. (simplex from B(1) to A(1)).

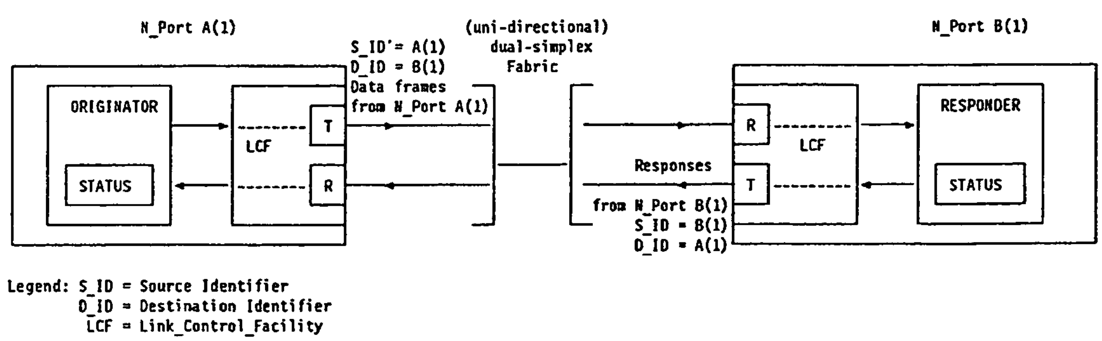


Figure L.3 - FC-2 dual-simplex communication model example

— Multiple Data frames from N_Port A(1) and the respective responses from N_Port B(1) results in similar flow.

Figure L.3 depicts the FC-2 dual-simplex communication model with its logical structure and components. The FC-2 dual-simplex communication model is functionally similar to the half-duplex model with the following difference. While the half-duplex communication model supports simultaneous bidirectional flow of frames, the dual-simplex model allows only unidirectional flow at any given time.

Annex M
(informative)
Bandwidth computation

M.1 Frame and inter-frame overhead

Frame and inter-frame overhead included in this standard is shown in table M.1.

Table M.1 - Frame overhead (bytes)					
Item	SOF	Frame Header	CRC	EOF	Total
Data Frame	4	24	4	4	36
ACK_1	4	24	4	4	36
6 Idles or R_RDY	NA	NA	NA	NA	24
NOTE - NA - Not Applicable					

M.2 Communication model overhead

FC-2 overhead varies with respect to communication model in use. Communication model overhead is shown in table M.2.

Table M.2 - Model overhead (bytes)			
Item	Model 1	Model 2	Model 3
Data Frame	36	36	36
ACK_1	NA	36	36
Idles or R_RDY	24	48	48
Total	60	120	120
NOTE - NA - Not Applicable			

M.3 Bandwidth computation examples

Bandwidth is a function of variants such as communication model, Fibre speed, and Payload size. Bandwidth computation examples of some choices are shown in table M.3. Bandwidth is given by:

$$Bandwidth(MB/s) = \frac{Payload\_size}{Payload\_size + overhead} x \frac{speed(Mbaud)}{10}$$

Table M.3 - Bandwidth computation examples					
Model	Speed	Payload_size (bytes)	Overhead (bytes)	Total (bytes)	Bandwidth (MB/sec)
1	1,0625 Gbaud	2 048	60	2 108	103,22 1)
1	531,25 Mbaud	2 048	60	2 108	51,61 1)
1	265,625 Mbaud	2 048	60	2 108	25,805 1)
2	1,0625 Gbaud	2 048	120	2 168	100,37 2)
2	531,25 Mbaud	2 048	120	2 168	50,185 2)
2	265,625 Mbaud	2 048	120	2 168	25,093 2)
3	1,0625 Gbaud	2 048	120	2 168	100,37 3)
3	531,25 Mbaud	2 048	120	2 168	50,185 3)
3	265,625 Mbaud	2 048	120	2 168	25,093 3)

NOTES
1) Aggregate for both Fibres
2) Per direction
3) Aggregate for all communicating N_Ports

Annex N
(informative)
CRC generation and checking

N.1 Extract from FDDI

First part of this annex is an extract from Fiber Distributed Data Interface (FDDI) Media Access Control (MAC) (Document reference: ISO/IEC 9314-2:1989). FDDI's Frame Check Sequence (FCS) methodology, polynomials and equations are used by Cyclic Redundancy Check (CRC) in FC-2. The term FCS is unique to FDDI and not used by Fibre Channel. Note that CRC coverage of FC-2 fields is specified in FC-2 part of FC-PH.

4.3.6 Frame check sequence (FCS)

This annex specifies the generation and checking of the FCS field. This field is used to detect erroneous data bits within the frame as well as erroneous addition or deletion of bits to the frame. The fields covered by the FCS field include the FC, DA, SA, INFO, and FCS fields.

4.3.6.1 Definitions

F(x) - A degree k-1 polynomial which is used to represent the k bits of the frame covered by the FCS sequence (see section 4.2.2). For the purposes of the FCS, the coefficient of the highest order term shall be the first bit transmitted.

L(x) - A degree 31 polynomial with all of the coefficients equal to one, ie.,

$$L(x) = X^{31} + X^{30} + X^{29} + \cdots + X^2 + X + 1$$

G(x) - The standard generator polynomial

$$G(x) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$$

R(x) - The remainder polynomial which is of degree less than 32
P(x) - The remainder polynomial on the receive checking side which is of degree less than 32
FCS - The FCS polynomial which is of degree less than 32
Q(x) - The greatest multiple of G(x) in $[X^{32}F(x)+X^{k}L(x)]$
$Q^{*}(x)$ - $X^{32}Q(x)$
M(x) - The sequence which is transmitted
$M^{*}(x)$ - The sequence which is received
C(x) - A unique polynomial remainder produced by the receiver upon reception of an error free sequence. This polynomial has the value

$$C(X)=X^{32}L(X)/G(X)$$

$$C(X) = X^{31} + X^{30} + X^{26} + X^{25} + X^{24} + X^{18} + X^{15} + X^{14} + X^{12} + X^{11} + X^{10} + X^{8} + X^{6} + X^{5} + X^{4} + X^{3} + X + 1$$

4.3.6.2 FCS generation equations

The equations which are used to generate the FCS sequence from F(x) are as follows:

$$FCS = L(X)+R(X) = R\$(X) \quad (1)$$

where R$(X) is the one's complement of R(X)

$$[X^{32}F(x)+X^{k}L(X)]/G(X)=Q(X)+R(X)/G(X) \quad (2)$$

$$M(x) = x^{32}F(x) + FCS \quad (3)$$

NOTE - All arithmetic is modulo 2.

In equation (1), note that adding L(x) (all ones) to R(x) simply produces the one's complement of R(x); this equation is specifying that the R(x) is inverted before it is sent out.

Equation (3) simply specifies that the FCS is appended to the end of F(x).

4.3.6.3 FCS checking

The received sequence $M^{*}(x)$ may differ from the transmitted sequence M(x) if there are transmission errors. The process of checking the sequence for validity involves dividing the received sequence by G(x) and testing the remainder. Direct division, however, does not yield a unique remainder because of the possibility of leading zeros. Thus a term L(x) is prepended to $M^{*}(x)$ before it is divided. Mathematically, the received checking is shown in equation (4).

$$X^{32}[M^{*}(X)+X^{k}L(X)]/G(X)=Q^{*}(X)+P(X)/G(X) \quad (4)$$

In the absence of rrors, the unique remainder is the remainder of the division

$$P(X)/G(X) = X^{32}L(X)/G(X) = C(X) \quad (5)$$

N.2 CRC g nerati n

The transmission bit order is defined as the bit ordering used by FC-1 prior to 8B/10B encoding (see clause 11), with bit A as the first bit transmitted and bit H as the last. The CRC field is calculated on the Frame_Header and the Data field content, using the defined transmission bit order (see 17.5), represented as the polynomial F(x) in figure N.1. The CRC field R$(x) is transmitted sequentially starting with the highest order term (X^{31}) and ending with the lowest order term (X^0).

The figure N.1 illustrates the flow from FC-2 to FC-0 using the FC-PH notation, including a reference to the mathematical notation used in N.1. The FC-2 word notation designates the most significant bit of the word as bit 31 and the least significant bit as 0. The FC-2 passes the words to FC-1 from left to right (ie., Word 0 through n). The FC-1 transmits each byte starting with the least significant bit, namely bits A-E first followed by bits F-H. For example, in figure N.1, the first byte of word 0 is transmitted with bit 24 first as bit A and bit 31 last as bit H. The order of the polynomial F(x) covered by the CRC and the order of the CRC polynomial R$(x) are illustrated in figure N.1.

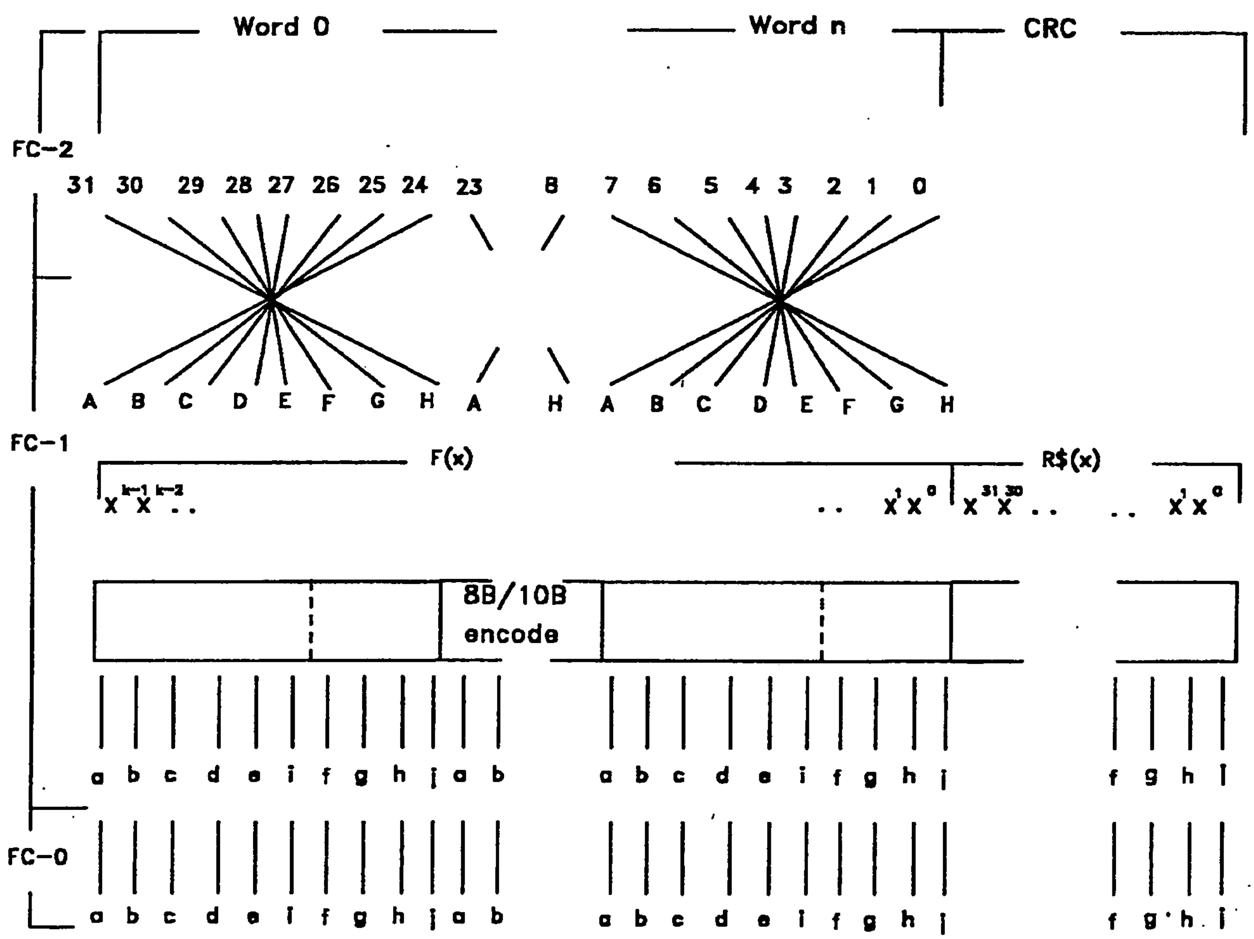


Figure N.1 - CRC coverage and bit ordering

N.3 Transmit rder of a word

The transmit order of an FC-2 word in the example illustrated in figure N.1 is expanded in figure N.2. The top half of the figure shows the ordering of the first halfword sent and the bottom half shows the ordering of the second halfword.

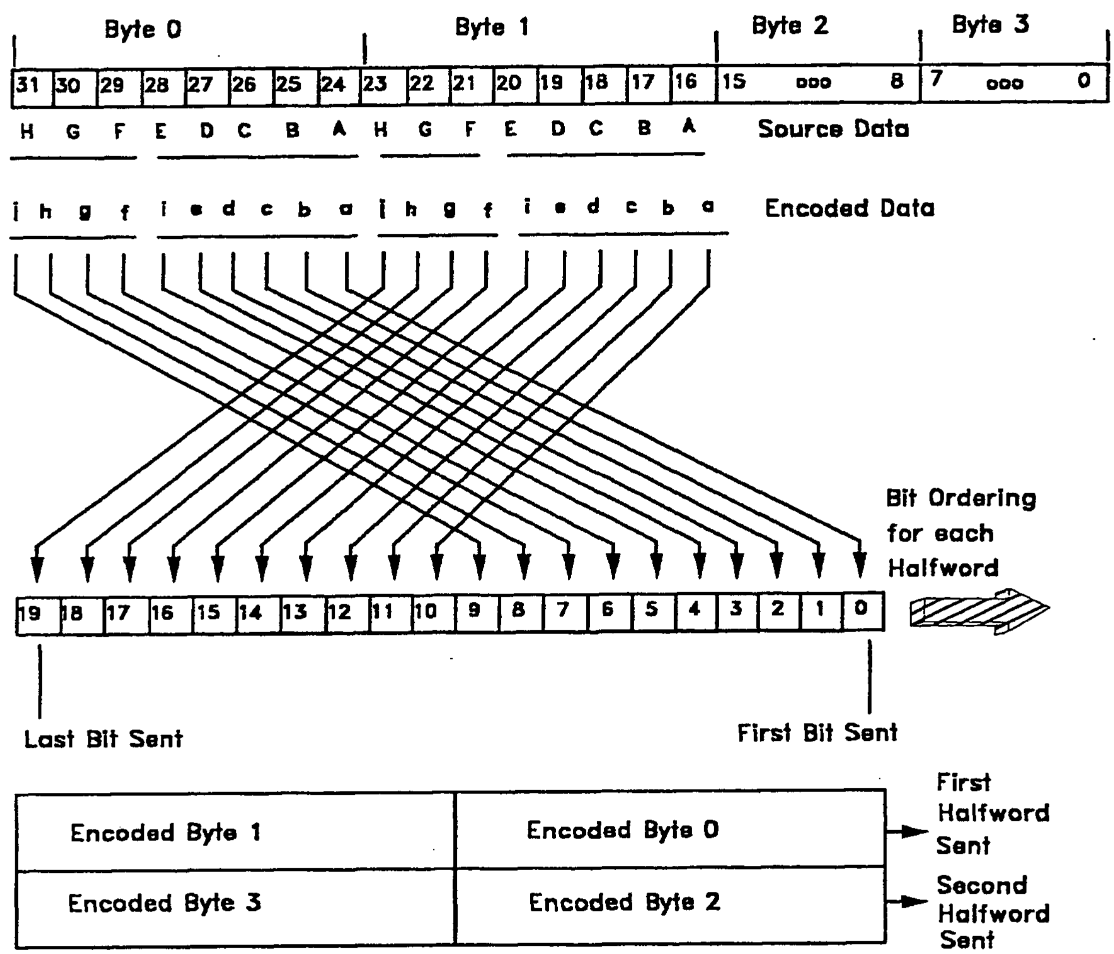


Figure N.2 - Word transmit order

N.4 CRC generation example for ACK_1 frame

An example of CRC generation for an ACK_1 frame is provided in a set of tables N.1 through N.8. Table N.1 shows an example of an ACK_1 fields without CRC and table N.2 shows the hexadecimal values for each field. Table N.3 shows the transmit bit order (038040C...080 hex) with the bytes in table N.2 transposed. Table N.4 shows the bit stream $X^{32}F(x)+X^{k}L(x)$ (FC7F...080 hex) for the sample. Table N.5 shows the generated remainder (649EOBF7 hex) for the sample. Table N.6 shows the one's complement of the remainder (9B61F408 hex) for the sample. The transmitted bit sequence for the sample with the CRC (038040C...F408 hex) is shown in table N.7. The transmitted 10B stream for the sample with CRC is shown in table N.8

Table N.1 - Sample FC-2 frame

Sample ACK_1 without CRC (frame header fields)		
R_CTL	D_ID	
rrrr rrrr	S_ID	
TYPE	F_CTL	
SEQ_ID	DF_CTL	SEQ_CNT
OX_ID		RX_ID
Parameter		

Table N.2 - Sampl ACK_1 with ut CRC

Sample Frame Header			
C0	01	02	03
00	04	05	06
00	C0	00	00
02	00	00	03
FF	FF	FF	FF
00	00	00	01

Table N.3 - F(x)

Bytes in table N.2 transposed			
03	80	40	C0
00	20	A0	60
00	03	00	00
40	00	00	C0
FF	FF	FF	FF
00	00	00	80

Table N.4 - X32 F(x) + X**k L(x)**

FC	7F	BF	3F
00	20	A0	60
00	03	00	00
40	00	00	C0
FF	FF	FF	FF
00	00	00	80

Table N.5 - R(x)

64	9E	0B	F7

Table N.6 - L(x) + R(x) = R$(x)

9B	61	F4	08

Table N.7 - M(x)

03	80	40	C0
00	20	A0	60
00	03	00	00
40	00	00	C0
FF	FF	FF	FF
00	00	00	80
9B	61	F4	08

Table N.8 - M(x) - (10B)

D0.6	D1.0	D2.0	D3.0
D0.0	D4.0	D5.0	D6.0
D0.0	D0.6	D0.0	D0.0
D2.0	D0.0	D0.0	D3.0
D31.7	D31.7	D31.7	D31.7
D0.0	D0.0	D0.0	D1.0
D25.6	D6.4	D15.1	D16.0

Annex O
(informative)
ACK_N Usage

Three examples (examples 1 through 3) on the usage of ACK_N and one example (example 4) on ACK_1 are shown. Figure O.1 illustrates a case where all Data frames are received in sequential order. Figures O.2, O.3 and O.4 correspond to cases when the Data frames are received out of order.

Order of SEQ_CNT of Data frames received	0	1	2	3	4	5	6	7	8	9	10	11	12	13 + End_SEQ = 1
ACK_N sent after receiving this frame				a				b				c		d
SEQ_CNT in ACK_N				3				7				11		13
History bit in Parameter field				0				0				0		0
ACK_CNT in Parameter field				4				4				4		2

ACK_N sent at	ACK_N content			SEQ_CNT of Data frames responded to
	Parameter field		Sequence Count field	
	History bit	ACK_CNT		
a	0	4	3	0,1,2 and 3
b	0	4	7	4,5,6 and 7
c	0	4	11	8,9,10 and 11
d	0	2	13	12 and 13

Figure O.1 - Exampl 1 - ACK_N usage

Order of SEQ_CNT of Data frames received	0	1	2	3	5	4	6	7	8	10	9	11	12	13 + End_SEQ = 1
ACK_N sent after receiving this frame				a				b				c		d
SEQ_CNT in ACK_N				3				7				11		13
History bit in Parameter field				0				0				0		0
ACK_CNT in Parameter field				4				4				4		2

ACK_N sent at	ACK_N content			SEQ_CNT of Data frames responded to
	Parameter field		Sequence Count field	
	History bit	ACK_CNT		
a	0	4	3	0,1,2 and 3
b	0	4	7	4,5,6 and 7
c	0	4	11	8,9,10 and 11
d	0	2	13	12 and 13

Figure O.2 - Example 2 - ACK_N usage

Order of SEQ_CNT of Data frames received	0	1	2	4	5	6	7	8	9	10	11	12	13	3 + End_SEQ = 1
ACK_N sent after receiving this frame				a			b				c			d e
SEQ_CNT in ACK_N				2			7				11			3 13
History bit in Parameter field				0			1				1			0 0
ACK_CNT in Parameter field				3			4				4			1 1

ACK_N sent at	ACK_N content			SEQ_CNT of Data frames responded to
	Parameter field		Sequence Count field	
	History bit	ACK_CNT		
a	0	3	2	0,1,2 and 3
b	1	4	7	4,5,6 and 7
c	1	4	11	8,9,10 and 11
d	0	1	3	3
e	0	1	13	13

Figure O.3 - Example 3 - ACK_N usage

Order of SEQ_CNT of Data frames received	0	1	2	4	5	6	7	8	9	10	11	12	13 + End_SEQ=1	3
ACK_N sent after receiving this frame	a	b	c	d	e	f	g	h	i	j	k	l		m n
SEQ_CNT in ACK_N	0	1	2	4	5	6	7	8	9	10	11	12		3 13
History bit in Parameter field	0	0	0	1	1	1	1	1	1	1	1	1		0 0
ACK_CNT in Parameter field	1	1	1	1	1	1	1	1	1	1	1	1		1 1

ACK_N sent at	ACK_N content			SEQ_CNT of Data frames responded to
	Parameter field		Sequence Count field	
	History bit	ACK_CNT		
a - c	0	1	0 - 2	0 - 2
d - l	1	1	4 - 12	4 - 12
m	0	1	3	3
n	0	1	13	13

Figure O.4 - Example 4 - ACK_1 usage

Ann x P
(informative)
Data transfer protocols and examples

This annex provides Data transfer protocol examples.

P.1 Frame level protocol

Class dependency of frame level protocol are shown.

P.1.1 Class 1 frame level protocol

Class 1 frame level protocol employs

a) Data frame
b) ACK
c) R_RDY (response to SOFc1)

Class 1 frame level protocol is illustrated in figure P.1.

a) The Sequence is initiated by the Originator with a Data frame embedded with SOFc1 (or SOFi1).
b) SOFc1 is used to indicate a Connect request and the Sequence initiation. This Data frame with SOFc1 is responded with a R_RDY by the F_Port and the destination N_Port, as illustrated. Next Data frame is not sent until the Connect request is accepted by the destination, unless the Fabric supports stacked connect-requests. (With stacked connect-requests, the only Data frame which can be sent before the connect-request is accepted by the destination ia another connect-request to a different destination.)
c) Within an established Connection SOFi1 indicates the Sequence initiation.
d) The Originator streams multiple Data frames and the Responder responds with ACK.
   - ACK returns following information contained in F_CTL of the Data frame to which it is responding, unaltered.
     - First Sequence
     - Last Sequence
     - Last Data frame
     - Sequence transmit initiative
   - ACK toggles following information contained in F_CTL of the Data frame.
     - Exchange Context
     - S quence Context

F_CTL usage for the Sequence is described in table P.1.

e) SOFn1 is used to indicate the Sequence in progress.
f) The end of Sequence is indicated to the Sequence Recipient by the Last Data frame bit in F_CTL. The last ACK is embedded with EOFt or EOFdt which indicates the end of Sequence to the Sequence Initiator.
g) The end of Exchange is indicated to the Sequence Recipient by the Last Sequence and the Last Data frame bits in F_CTL. The Last Sequence and the EOFt or EOFdt indicate to either N_Port the end of Exchange.

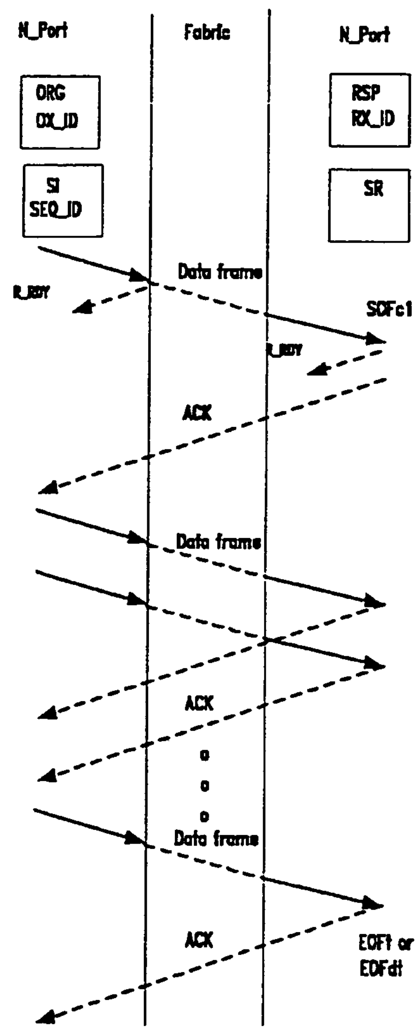


Figure P.1 - Class 1 frame level protocol

P.1.2 Class 2 frame level pr t col

Class 2 frame level protocol employs

a) Data frame
b) ACK
c) R_RDY

Class 2 frame level protocol is illustrated in figure P.2.

a) The Sequence is initiated by the Originator with a Data frame embedded with SOFi2.
b) The F_Port responds with an R_RDY and forwards the Data frame to the destination.
c) The destination responds with an R_RDY, in addition to ACK.
d) The F_Port and the N_Port respond each with R_RDY on receipt of ACK.
e) The Originator streams multiple Data frames and the Responder responds with ACK.
   - ACK returns some information contained in F_CTL of the Data frame to which it is responding unaltered.
     - First Sequence
     - Last Sequence
     - Last Data frame
     - Sequence transmit initiative
   - ACK toggles some information contained in F_CTL of the Data frame.
     - Exchange Context
     - Sequence Context

   F_CTL usage for the Sequence is described in table P.1.
f) For each of these frames received, each N_Port or F_Port returns a R_RDY.
g) SOFn2 is used to indicate the Sequence in progress.
h) The end of Sequence is indicated by the Sequence Initiator by the Last Data frame bit in F_CTL. However, the Sequence ends in the perspective of Sequence Recipient, only when all Data frames are received or accounted for.
i) The Sequence Recipient transmits EOFt only in the final ACK after all Data frames are received or accounted for.

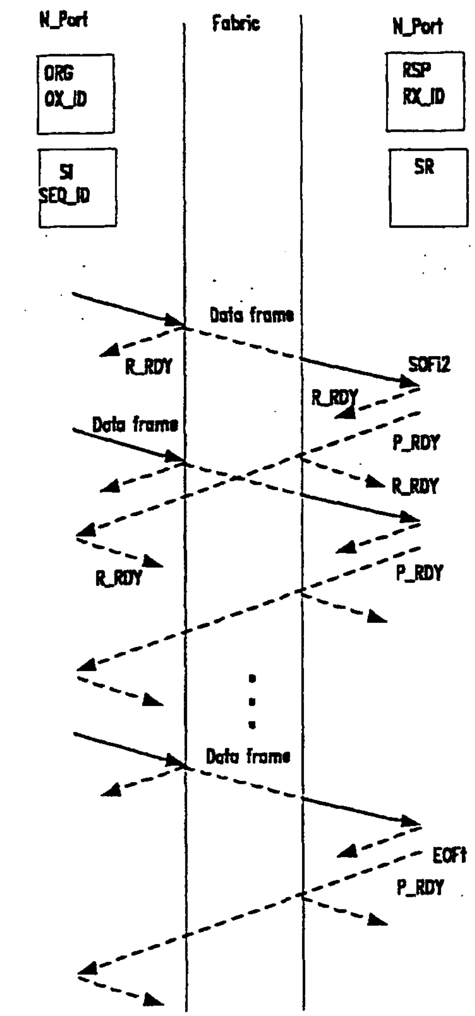


Figure P.2 - Class 2 frame level protocol

P.1.3 Class 3 frame level pr toc l

Class 3 frame level protocol employs

a) Data frame
b) R_RDY

Class 3 frame level protocol is illustrated in figure P.3.

a) The Sequence is initiated by the Originator with a Data frame embedded with SOFi3.
b) The F_Port responds with an R_RDY and forwards the Data frame to the destination.
c) The destination responds with an R_RDY.
d) The Originator streams multiple Data frames. For each of these frames received, each N_Port or F_Port returns a R_RDY. F_CTL usage for the Sequence is described in table P.2.
e) SOFn3 is used to indicate the Sequence in progress.
f) The end of Sequence is indicated to the Sequence Recipient by the Last Data frame bit in F_CTL and EOFt.

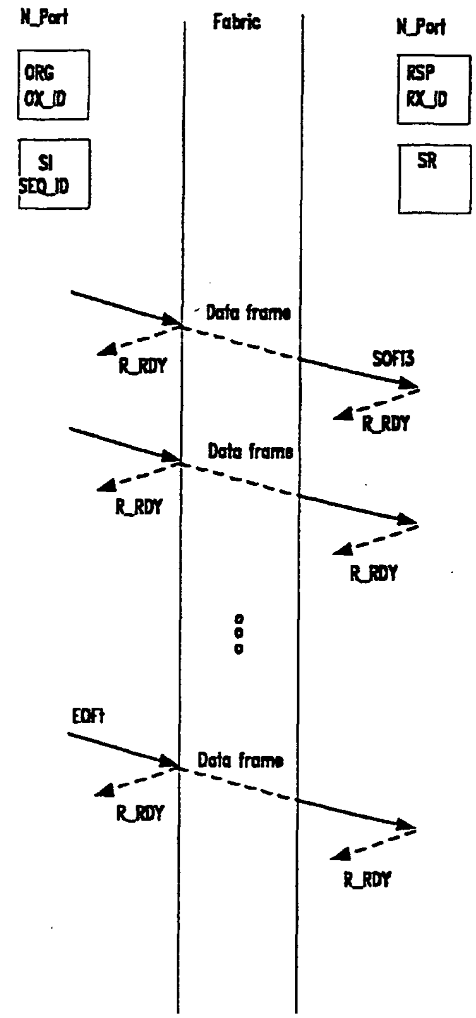


Figure P.3 - Class 3 frame level protocol

Table P.1 - F_CTL for Class 1 or Class 2 frame level protoc l						
Description	**Exchange Context**	**Sequence Context**	**First Sequence of Exchange**	**Last Sequence of Exchange**	**End Sequence**	**Sequence transmit initiative**
F_CTL Bits	23	22	21	20	19	16
First Data frame	0 (ORG)	0 (SI)	1 (First	0 Sequence)	0	0 (NM)
ACK	1 (RSP)	1 (SR)	1 (First	0 Sequence)	0	0 (NM)
Interme-diate Data frame(s)	0	0	1	0	0	0 (NM)
ACK	1	1	1	0	0	0 (NM)
Last Data frame	0	0	1	0	1	0 (retain Sequence Initiative)
ACK	1	1	1	0	1	0 (NM)
Note: NM - Not Meaningful						

Table P.2 - F_CTL for Class 3 frame level protocol						
Description	**Exchange Context**	**Sequence Context**	**First Sequence of Exchange**	**Last Sequence of Exchange**	**Last Data frame of Sequence**	**Sequence transmit initiative**
F_CTL Bits	23	22	21	20	19	16
First Data frame	0 (ORG)	0 (SI)	1 (First	0 Sequence)	0	0 (NM)
Interme-diate Data frame	0	0	1	0	0	0 (NM)
Last Data frame	0	0	1	0	1	0 (retain Sequence Initiative)
Note: NM - Not Meaningful						

P.2 Sequence level protocol example

Sequence level protocol is illustrated with a three Sequence Exchange in figure P.4. The first Sequence is a "read" request. The second Sequence transfers the "data". The third Sequence transfers "ending status" and ends the Exchange.

Frames 1,2, and 3 represent the first Sequence of an Exchange. In this example a Command Request for a Read operation is sent as a request Sequence. Note that Sequence Initiative is transferred to the Sequence Recipient.

Frames 4,5 and 6 represent the first, intermediate and last frames of the data transferred in response to the Read request. Note that the Sequence Initiative is retained in order to start a Sequence with ending status.

Frames 7,8 and 9 represent the ending status for the preceding data transfer and end the Exchange. Depending on the Upper Level Protocol, the Responder may not be allowed to end the Exchange, but transfer the Sequence initiative to the Originator to complete the Exchange.

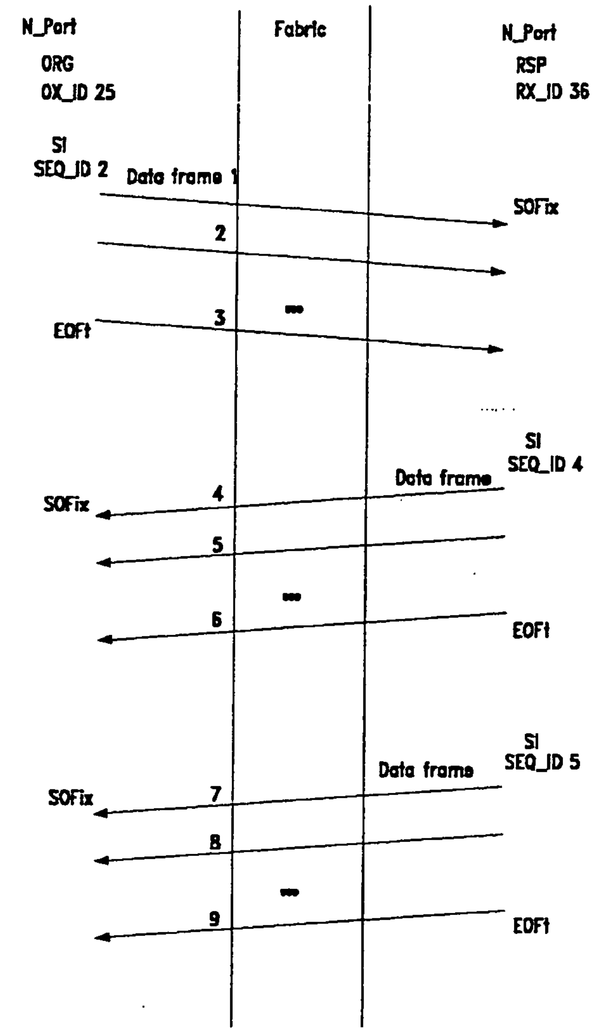


Figure P.4 - Sequence level protocol example

F_CTL usage

Use of F_CTL bits for these example Sequences are shown in table P.3.

Tabl P.3 - F_CTL for bidirectional Exchange xample						
Description	**Exchange Context**	**Sequence Context**	**First Sequence of Exchange**	**Last Sequence of Exchange**	**Last Data frame of Sequence**	**Sequence transmit initi-ative**
F_CTL Bits	23	22	21	20	19	16
1. First Data frame (SOFix) of the Exchange and of the first Sequence (a Read Request Sequence)	0	0	1	0	0	0 (NM)
2. Intermediate Data frame of first Sequence	0	0	1	0	0	0 (NM)
3. Last Data frame of first Sequence	0	0	1	0	1	1
4. First Data frame (SOFix) of intermediate Sequence (Reply Sequence)	1	0	0	0	0	0 (NM)
5. Intermediate Data frame of intermediate Sequence	1	0	0	0	0	0 (NM)
6. Last Data frame of intermediate Sequence	1	0	0	0	1	0
7. First Data frame (SOFix) of the Last Sequence (Reply Status Sequence)	1	0	0	1	0	0 (NM)
8. Intermediate Data frame of the Last Sequence	1	0	0	1	0	0 (NM)
9. Last Data frame of the Last Sequence and of the Exchange	1	0	0	1	1	0

P.3 Class 1 frame lev l prot col example

N_Port Login is used as an example to illustrate Class 1 frame flow.

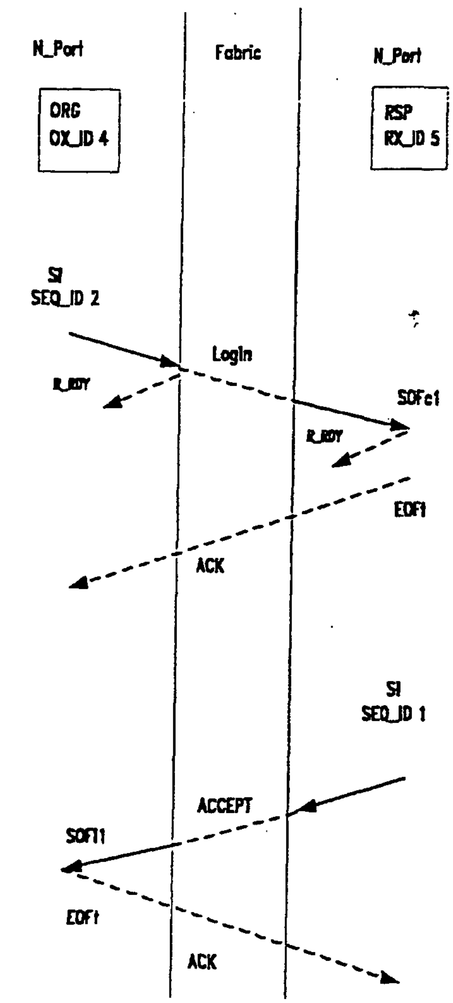


Figure P.5 - Class 1 frame level protocol - Login example

P.4 Class 2 frame level protocol exampl

N_Port Login is used to illustrate Class 2 frame flow.

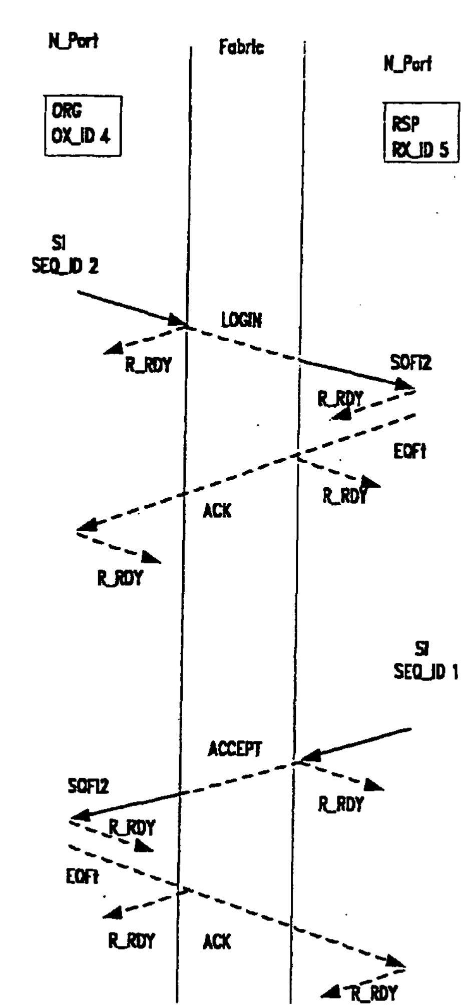


Figure P.6 - Class 2 frame level protocol - Login example

P.5 Class 3 frame level pr tocol example

N_Port Login is used to illustrate Class 3 frame flow.

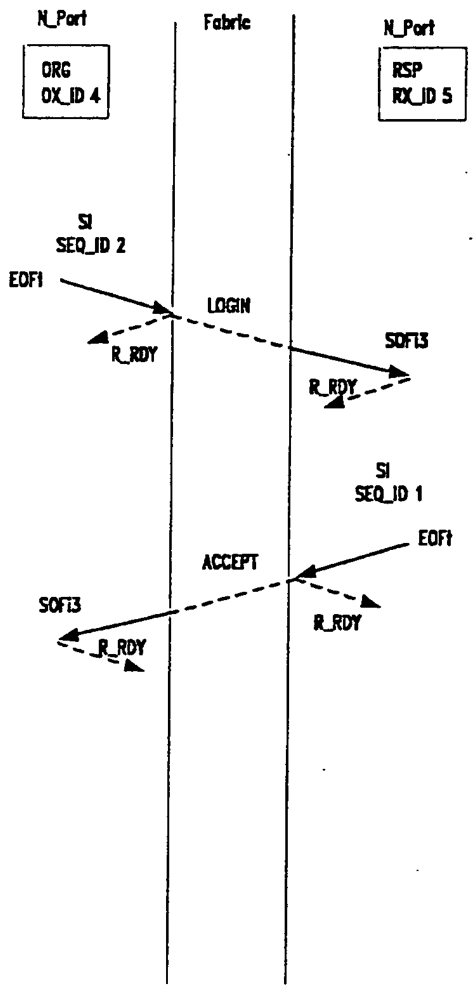


Figure P.7 - Class 3 frame level protocol - Login example

Annex Q
(informative)
Connection management applications

Q.1 Example cases

The following Cases represent examples of how a Class 1 Connection is removed using the Remove Connection procedure.

Q.1.1 Case 1

Table Q.1 shows a Case where N_Port (A) transmits its last Sequence without N_Port (B) transmitting any Sequences.

Table Q.1 - Case 1

xmit E_C (A)	recv E_C (A)	N_Port Actions A = this N_Port B = other connected N_Port
0	0	A transmitting Sequences
1	0	A transmits last Data frame of last Sequence
0	0	A receives the ACK_1 or ACK_N to the last Data frame with **EOFdt** -- Connection Removed

Q.1.2 Case 2

Table Q.2 shows a Case where N_Port (A) completes its last Sequence before N_Port (B).

Table Q.2 - Case 2

xmit E_C (A)	recv E_C (A)	N_Port Actions A = this N_Port B = other connected N_Port
0	0	Both A and B transmitting Sequences
1	0	A transmits last Data frame of last Sequence
0	0	B completes Sequences in progress, A responds with Link_Continues
0	1	A receives the last Data frame of last Sequence from B
0	0	A transmits **EOFdt** on ACK_1/ACK_N -- Connection Removed

Q.1.3 Case 3

Table Q.3 shows a Case where N_Port (B) transmits its last Sequence before N_Port (A).

Table Q.3 - Case 3

xmit E_C (A)	recv E_C (A)	N_Port Actions A = this N_Port B = other connected N_Port
0	0	Both A and B transmitting Sequences
0	1	A receives last Data frame of last Sequence from B
0	0	A completes Sequences in progress
1	0	A transmits last Data frame of last Sequence
0	0	A receives EOF_{dt} on ACK_1/ACK_N -- Connection Removed

Q.1.4 Case 4

Table Q.4 shows a Case where N_Port (A) transmits its last Data frame of its last Sequence simultaneously with receiving the last Data frame of N_Port (B)'s last Sequence.

Table Q.4 - Case 4

xmit E_C (A)	recv E_C (A)	N_Port Actions A = this N_Port B = other connected N_Port
0	0	Both A and B transmitting Sequences (A is Connection Initiator)
1	0	A transmits last Data frame of last Sequence
0	1	A receives Data frame from B before receiving Link_Continue for its last Data frame with E_C = 1
0	0	A waits for its Link_Continue response
0	0	A receives its ACK_1/ACK_N from B with EOF_{n1}
0	0	A transmits ACK_1/ACK_N with EOF_{dt} -- Connection Removed

Q.2 Ending sequ nce and Connecti n

Table Q.5 shows an example of F_CTL Bit settings for key Data frames within an example Exchange.

The first Sequence is a "read" request. The second Sequence transfers the "data". The third Sequence transfers "ending status" and ends the Exchange and terminates a Class 1 Dedicated Connection.

Table Q.5 - F_CTL for example Exchange

Description	Exc Owner	Seq Owner	First Seq of Exc	Last Seq of Exc	Last Seq Data frame	E_C	Seq xmit initiative
Bits	23	22	21	20	19	18	16
1.First Data frame of Exchange (SOFix) - a Read request(xmit)	0	0	1	0	0	0	0 (NM)
2.Intermediate Data frame of first Sequence (xmit)	0	0	1	0	0	0	0 (NM)
3.Last Data frame of first Sequence (xmit)	0	0	1	0	1	0	1
4.First Data frame of reply "data" Sequence (SOFix)(recv)	1	0	0	0	0	0	0 (NM)
5.Last Data frame of reply data Sequence (recv)	1	0	0	0	1	0	0
6.First Data frame of reply Status Sequence (SOFix)(recv)	1	0	0	1	0	0	0 (NM)
7.Last Data frame of Exchange (recv)	1	0	0	1	1	1	1
8.Last ACK_1 of Exchange (xmit) with EOFdt	0	1	0	1	0	0	0

Frames 1,2, and 3 represent the first Sequence of an Exchange. In this example a Command Request for a Read operation is sent as a request Sequence. Note that Sequence Initiative is transferred to the Sequence Recipient.

Frames 4 and 5 represent the first and last frames of the data transfer associated with the Read operation. Note that the Sequence Initiative is retained in order to start a Sequence with ending status.

Frames 6 and 7 represent the ending status for the preceding data transfer and ends the Exchange. Frame 7 ends the Sequence, ends the Exchange, and requests termination of the Connection. The ACK_1 response to frame 7 (frame 8) removes the Dedicated Connection with **EOFdt**.

Q.3 Port states

A Port's behavior relative to its Receiver and Transmitter may be viewed as a number of separate and distinct states within a Port with respect to Primitive Sequences.

Active (AC)

A Port is in the Active state when it is not in any other states. In the Active state, a Port is operational and available to transmit and receive frames. The Port is transmitting and receiving Idles.

Link Recovery

When an N_Port is performing any portion of Link Recovery, it is considered "Busy" to other N_Ports attached to the Fabric, if present. Three substates exist during the Link Recovery State:

- **LR Transmit (LR1)**

  The Port transmits the LR Primitive Sequence to reset the attached Port or F_Port.

- **LR Receive (LR2)**

  The Port transmits the LRR Primitive Sequence as a result of receiving the LR Primitive Sequence.

- **LRR Receive (LR3)**

  The Port transmits Idles as a result of receiving the LRR Primitive Sequence.

Link Failure

When an N_Port is in the Link Failure state, the N_Port is unable to process frames. The Link Failure state is an abnormal occurrence at the N_Port and the N_Port is considered "not available" to other N_Ports attached to the Fabric, if present.

Two substates exist:

- **NOS Receive (LF1)**

  When NOS is received, the Port transmits OLS.

- **NOS Transmit (LF2)**

  When a Port is not operational, it transmits NOS to indicate a Link failure.

Offline

The Offline state may be entered in response to receiving OLS on its link, or by action initiated by the link control facility within the Port.. The link control facility enters the Offline state for three reasons:

a) during initialization, to proceed from Not_Operational to Operational such as at power-up,
b) during shutdown, to remove a Port from service, or
c) to perform Port diagnostics.
d) to indicate an internal failure.

Three substates exist:

- **OLS Transmit (OL1)**

  The Port transmits OLS. After the Port receives the LR Primitive Sequence or exceeds a timeout period, it may perform any link activity including turning off its transmitter without errors being detected by the other attached Port. The Port exits this state by receiving LR and transmitting the LRR Primitive Sequence.

- **OLS Receive (OL2)**

  When a Port receives OLS, it transmits the LR Primitve Sequence. It ignores link errors or other Primitive Sequences excluding LR or LRR. When the Port receives LR or LRR, it exits this substate.

- **Wait for OLS (OL3)**

  When a Port is in either OLS Transmit or OLS Receive State and it detects Loss of Signal or Loss of Synchronization for a period of time greater than a timeout period, the Port enters the Wait for OLS State and is Offline. It normally exits this state upon reception of OLS.

Q.4 State change table

Table Q.6 describes the state changes made based on the current state and the Primitive Sequence or Signal received. See 16.4.3 for specific additional considerations and conditions for entering and exiting the Offline State (OL1 or OL2).

Table Q.6 - N_Port states - Primitive Sequences									
STATE	ACTIVE	LINK RECOVERY			LINK FAILURE		OFFLINE		
INPUT EVENT SUBSTATE XMIT >>	IDLE RECV (AC) IDLE	LR XMIT (LR1) LR	LR RECV (LR2) LRR	LRR RECV (LR3) IDLE	NOS RECV (LF1) OLS	NOS XMIT (LF2) NOS	OLS XMIT (OL1) OLS	OLS RECV (OL2) LR	WAIT OLS (OL3) NOS
1. L >> LR	LR2	LR2	**	LR2	LR2	**	LR2 Note 2	LR2	LF2
2. L >> LRR	LR3 Note 3	LR3	LR3	**	**	**	**	LR3	LF2
3. L>> IDLES	**	**	AC	AC	**	**	**	**	**
4. L >> OLS	OL2	OL2	OL2	OL2	OL2	OL2	OL2 Note 2	**	OL2
5. L >> NOS	LF1	LF1	LF1	LF1	**	LF1	LF1 Note 2	LF1	LF1
6. Loss of Signal	LF2	LF2	LF2	LF2	LF2	** Note 1	OL3 Note 2	OL3 Note 1	**
7. Loss of Sync > Limit	LF2	LF2	LF2	LF2	LF2	**	OL3 Note 2	OL3	**
8. Event timeout (R_T_TOV)	N/A	LF2	LF2	LF2	LF2 Note 4	N/A	OL3 Note 2 Note 6	OL3 Note 5	N/A

LEGEND

L >> means receiving from the Link,

N/A means not applicable,

An ** entry means no change in state

NOTES

1 Depending of Laser safety requirements, the transmitter may enter a "pulse" transmission mode of operation when Loss of Signal is detected.

2 All events are ignored until the Port determines it is time to leave the OLS transmission state.

3 A Primitive Sequence Protocol error is detected (An improper Primitive Sequence was received in this State) The Primitive Sequence Protocol error count in the LESB is incremented.

4 The timeout period starts timing when NOS is no longer recognized and none of the other events occur which cause a transition out of the state.

5 The timeout period starts timing when OLS is no longer recognized and none of the other events occur which cause a transition out of the state.

6 The timeout period starts timing when the Port is attempting to go online, transmits OLS, and none of the other events occur which cause a transition out of the state.

L>>LR

The Port is receiving and recognizing the Link Reset Primitive Sequence.

L>>LRR

The Port is receiving and recognizing the Link Reset Response Primitive Sequence.

L>>Idles

The Port is receiving and recognizing Idles.

L>>OLS

The Port is receiving and recognizing the Offline Primitive Sequence.

L>>NOS

The Port is receiving and recognizing the Not Operational Primitive Sequence.

Loss of Signal

The Port has detected loss of signal.

Loss of Sync > Limit

The Port has detected loss of synchronization for a period of time greater than a timeout period.

Event Timeout (R_T_TOV)

The timeout for an expected event exceeds the Receive_Transmitter Timeout Value (R_T_TOV).

Annex R
(informative)
Association_Header examples

R.1 X_ID Invalidation (Operation_ Associator) example

The X_ID reassignment protocol allows for the efficient management of N_Port resources. This annex provides an example of how this protocol would be used for an operation between an IBM ES/9000[8] system and a disk.

R.1.1 Fibre Channel mapping

In this example, FC-2 constructs are mapped as follows:

a) Operation_Associator is mapped as subchannel identifier.

b) X_ID is mapped as Local State Buffer slot.

R.1.2 Background

The channel subsystem of an ES/9000 system performs operations with a cooperating process/device. An operation consists of the execution of a channel program, from initiation to termination. Each channel program consists of one or more Channel Command Words (CCWs) which describe the function(s) to be performed by a target process/device.

There may be points in time during an operation where the channel is not being actively used, e.g., a device is processing a request and determining what data to transfer. During this period, it is desirable to "logically disconnect" the device from the channel to free up channel resources for use with another operation. Similarly, when a channel is actively being used,

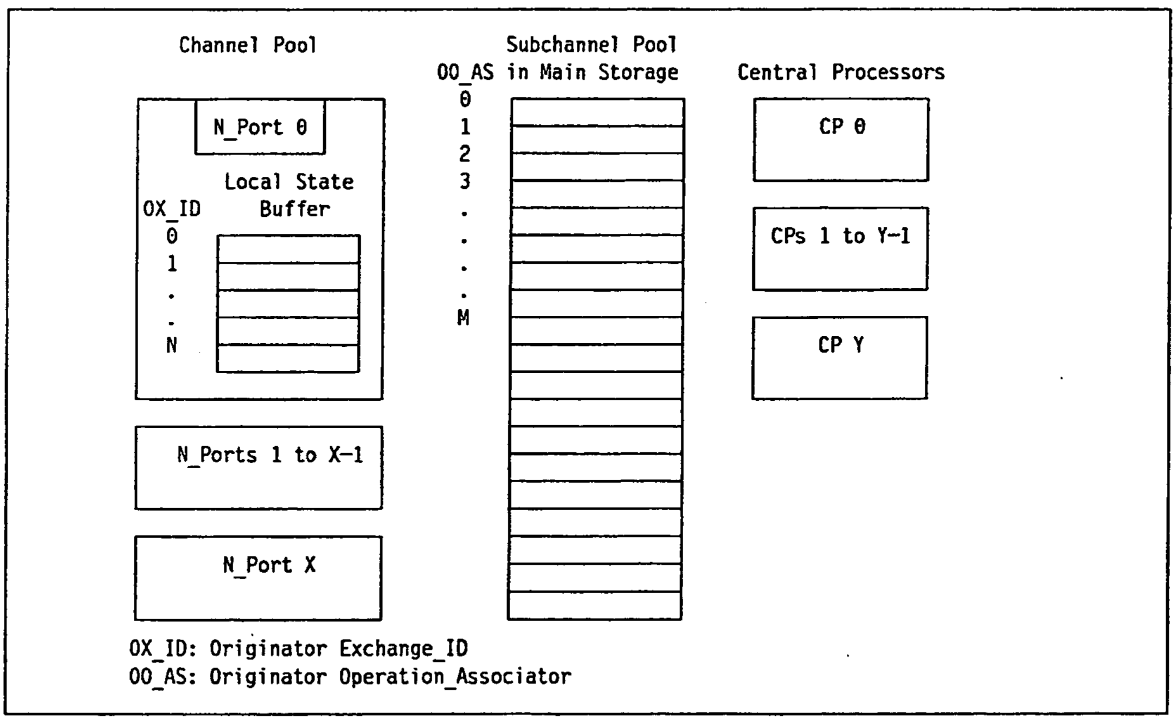


Figure R.1 - ES/9000 system structure

[8] ES/9000 is a trade-name of a product supplied by the International Business Machines Corporation. This information is given for the convenience of the users of this standard and does not constitute an endorsement by ANSI of the product named.

e.g., Data frames are being transmitted or received, the device is "logically connected" to the channel Since "logical connections" are tied to the use of N_Port resources, they are independent of Service Class. However, Class 1 service requires that a Dedicated connection exist in order for a logical connection to exist.

The structure of the ES/9000 channel subsystem is illustrated in figure R.1 and described below:

– A control block, called a "subchannel" is defined for each process or device with which the system is to communicate.

– Each subchannel manages one operation at a time for its associated process or device.

– A pool of subchannels is maintained in main storage.

– Every subchannel is accessible by every channel (N_Port) and every processor in the system.

– The subchannels are long-lived (across many distinct operations).

– A construct, called a "Local State Buffer", is maintained within each channel.

– The Local State Buffer consists of slots that contain the state information (including subchannel information) for each operation actively being executed by the channel.

– The information within a slot is relatively short-lived. It exists only while the channel is actively executing the operation, i.e., while a logical connection exists.

The example disk operation in the following sections is divided into several segments:

a) Initiation: A Processor has executed the instruction (Start Subchannel) that initiates the execution of a channel program. The following occurs:

1) An available channel having connectivity to the target disk is selected and the corresponding S_ID is generated.

2) A D_ID associated with the disk is selected.

3) A Local State Buff r slot in the selected channel is allocated and subchannel information is loaded from main storage into the slot.

4) An OX_ID is generated based on the allocated slot.

b) Disconnection: When the disk has reached a point in the operation where the channel is not required for a period of time, it logically disconnects from the channel to allow channel resources to be used by another device. For Class 1 service, this logical disconnection may cause a Dedicated Connection to be removed, depending upon other activity between the communicating N_Ports. The following occurs:

1) The information in the Local State Buffer slot is loaded back into the subchannel, which indicates that the operation is still active.

2) The Local State Buffer slot, and, hence, the X_ID, is available for use by another operation.

c) Reconnection: When the disk has reached a point in the operation where the channel is required, it logically reconnects to the channel. For Class 1 service, this reconnection may cause a Dedicated Connection to be established, depending upon other activity between the communicating N_Ports. The following occurs:

1) An available disk port having connectivity to the ES/9000 system is selected and the corresponding S_ID is generated.

2) A D_ID associated with the ES/9000 system is selected.

3) Upon receiving the reconnection request, a Local State Buffer slot in the channel corresponding to the D_ID is allocated and subchannel information is loaded into the slot from main storage.

NOTE - The selected channel used may be different than the channel selected during operation initiation.

d) Termination: When the disk has completed the I/O operation, it logically disconnects from the channel, indicating that this is the end of the operation. The following occurs:

1) The information in the Local State Buffer slot is loaded back into the subchannel, which indicates that the operation has ended.

2) The Local State Buffer slot, and, hence, the X_ID, is available for use by another operation.

The following sections detail how the X_ID reassignment protocol is used to perform the above functions.

R.1.3 Operation initiation

When an I/O operation is initiated with a disk device, the following occurs, as illustrated in figure R.2:

a) An N_Port pair (S_ID/D_ID) is selected for the operation.

b) The subchannel for the target process is "locked" and fetched into a slot in the channel's local state buffer. No other channel may now access this subchannel.

c) The X_IDs are set:

— The OX_ID is assigned for this operation, based upon the slot assigned in the local state buffer (e.g., OX_ID = 1).

— The RX_ID is set to nulls since it is not yet known

d) The Association_Header is initialized:

— The Originator Process_Associator is set as indicated in R.2.

— The Responder Process_Associator is set as indicated in R 2.

— The Originator Operation_Associator is set to indicate the subchannel (e.g., OO_AS = 3).

— The Responder Operation_Associator is set to nulls since it is not yet known.

e) The "New X_ID Assigned" bit is set for the Frame header.

f) The first Data frame of the operation is transmitted to the disk and the channel waits for the resultant ACK before sending any subsequent frames (which will not contain the Association_Header).

g) When the disk receives this Data frame, it will.

— Remember the Originator Operation_Associator and Process_Associator for the duration of the operation.

— Generate an RX_ID (e.g., 72) and return it in the ACK to this Data frame.

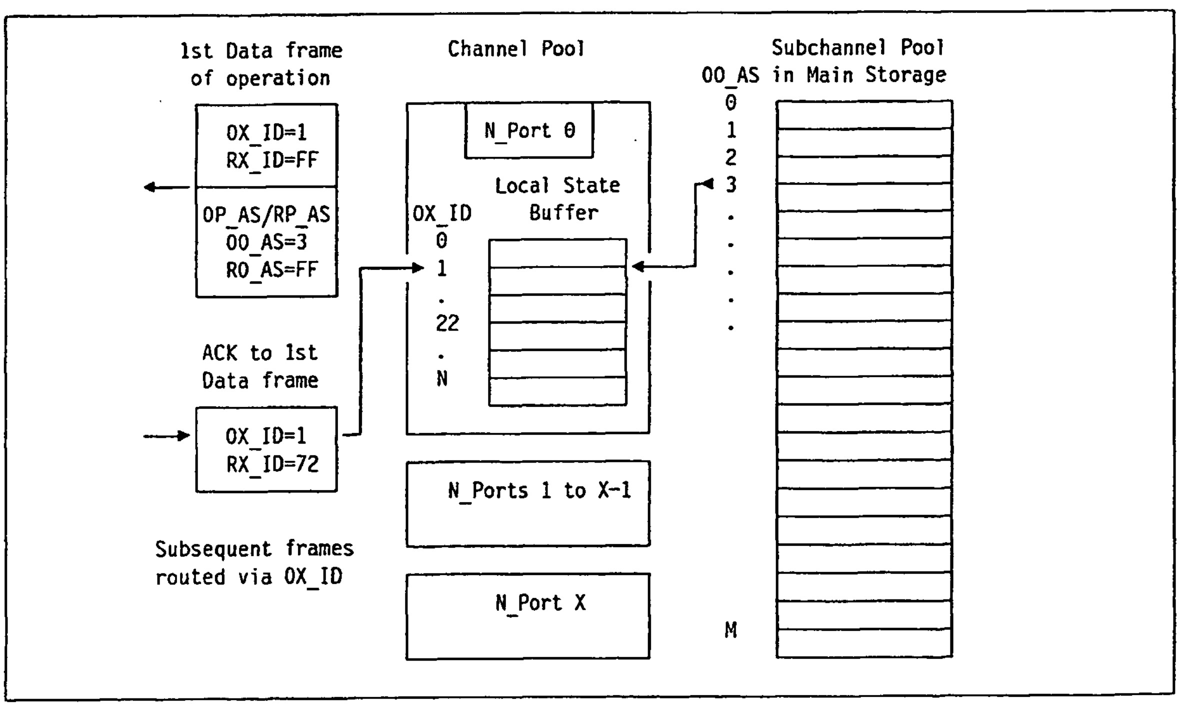


Figure R.2 - Operation initiation

h) At this point, the channel may continue the operation.

i) All frames received by the channel will be routed based on the OX_ID.

R.1.4 Operation disconnection

When the disk gets to a point in the operation where it does not need the channel, e g., processing of a Locate Record command, the following occurs, as illustrated in figure R.3:

a) To invalidate its RX_ID, it sets the "Invalidate X_ID" bit in the Frame header of the Data frame it returns to the channel.

b) If this is the only active Exchange and a Dedicated Connection exists, it will be terminated using the E_C protocol.

c) The channel sets the "Invalidate X_ID" bit in the frame header of the ACK frame returned in response to the last Data frame from the disk.

d) The channel stores the information from the Local State Buffer back into the associated subchannel in Main Storage.

e) This Local State Buffer slot and associated X_ID are now available for us by another Exchange.

R.1.5 Operation reconnection

When the operation is to be resumed (e g., the disk is ready for data transfer), the following occurs, as illustrated in figure R.4.

a) The disk selects an N_Port pair.

b) The X_IDs are set:

– The OX_ID is set to nulls, since the channel had indicated that it invalidated its X_ID.

– The RX_ID is set for this operation. Since the disk previously invalidated its X_ID, the "New X_ID assigned" bit is also set.

c) The Association_Header is initialized:

– The Originator and Responder Process_Associators are not meaningful in this example.

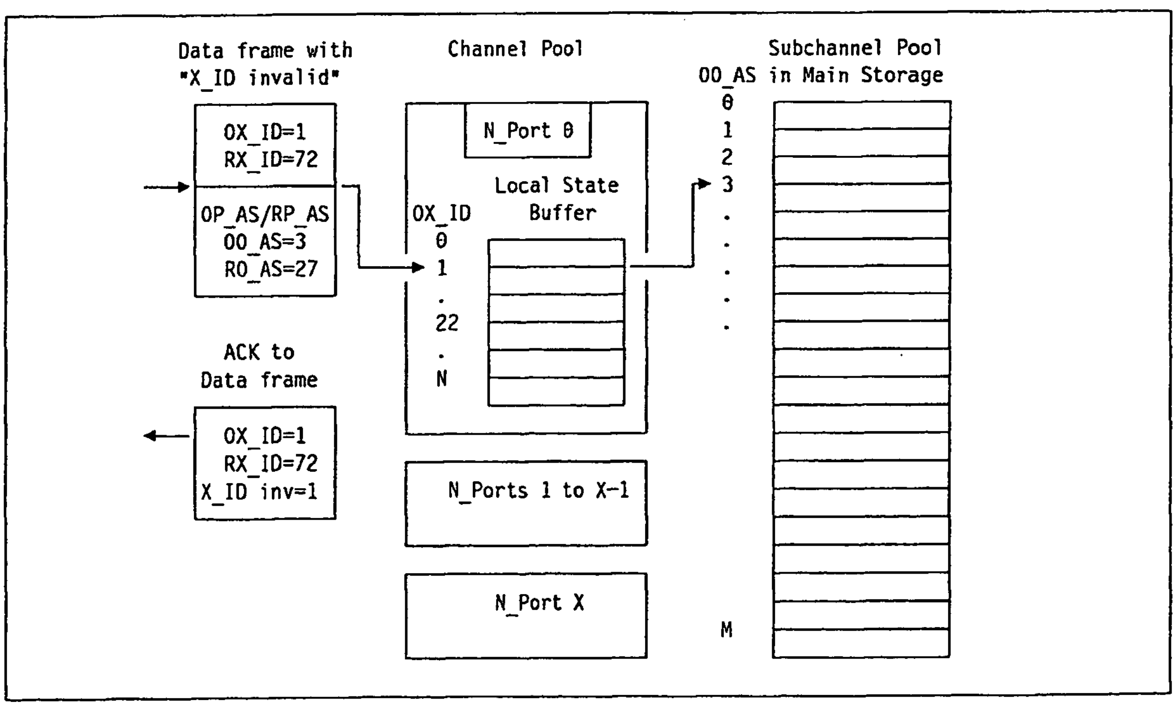


Figure R.3 - Operation disconnection

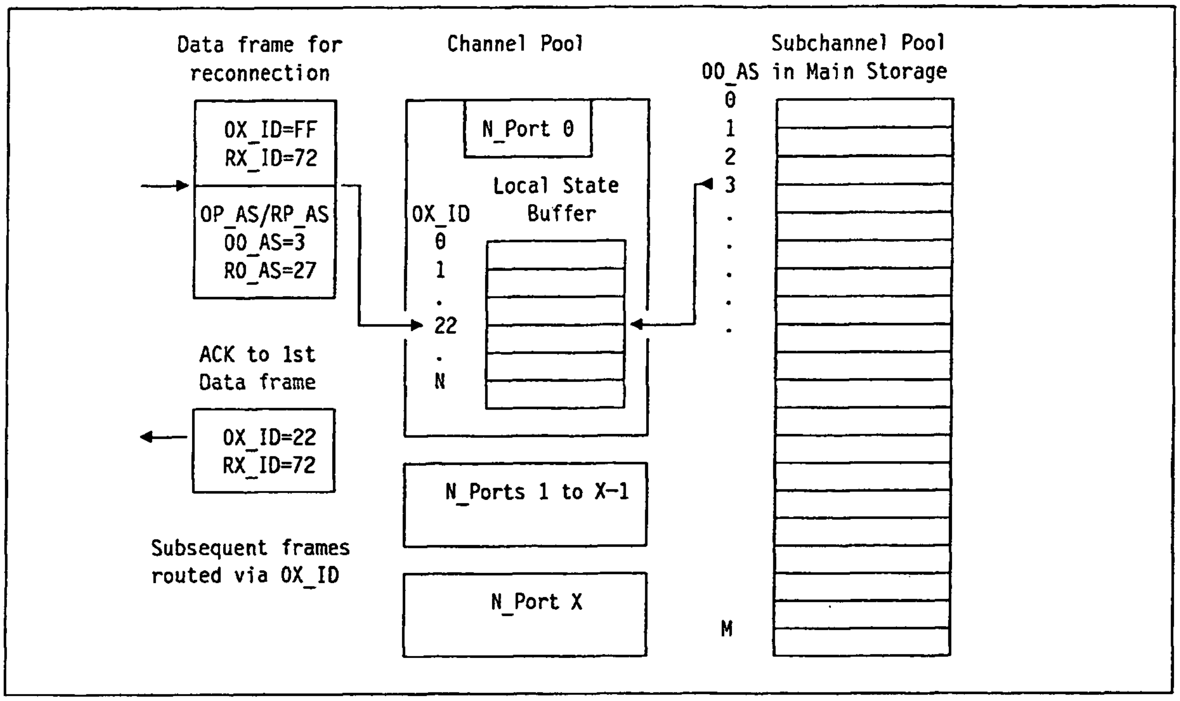


Figure R.4 - Operation reconnection

– The Originator Operation_Associator is set to the value in the Association_Header received at the start of the operation.

NOTE - The channel will use this value to locate the subchannel for this operation.

– The Responder Operation_Associator is set to whatever the disk wants.

d) The first Data frame of the reconnection is transmitted to a channel (note that this may be a different channel than originally initiated the operation), and the disk waits for the resultant ACK before sending any subsequent Data frames for this Exchange. The subsequent frames do not contain the Association_Header.

e) When the channel receives this Data frame, it will route the frame based on the Association_Header:

– Assign a slot in its Local State Buffer.

– Locate the subchannel for this operation from the Originator Operation_Associator (e g., OO_AS = 3).

– "Lock" this subchannel and fetch it into the assigned Local State Buffer slot.

– Generate an OX_ID (e.g., OX_ID = 22) and return it in the ACK to this Data frame.

f) All subsequent frames received by the channel for this Exchange are routed via the OX_ID.

R.1.6 Operation termination

When the operation is to terminated, the following occurs, as illustrated in figure R.5:

– The channel stores the information from the Local State Buffer back into the associated subchannel in Main Storage.

– This Local State Buffer slot and associated X_ID are now available for use by another Exchange.

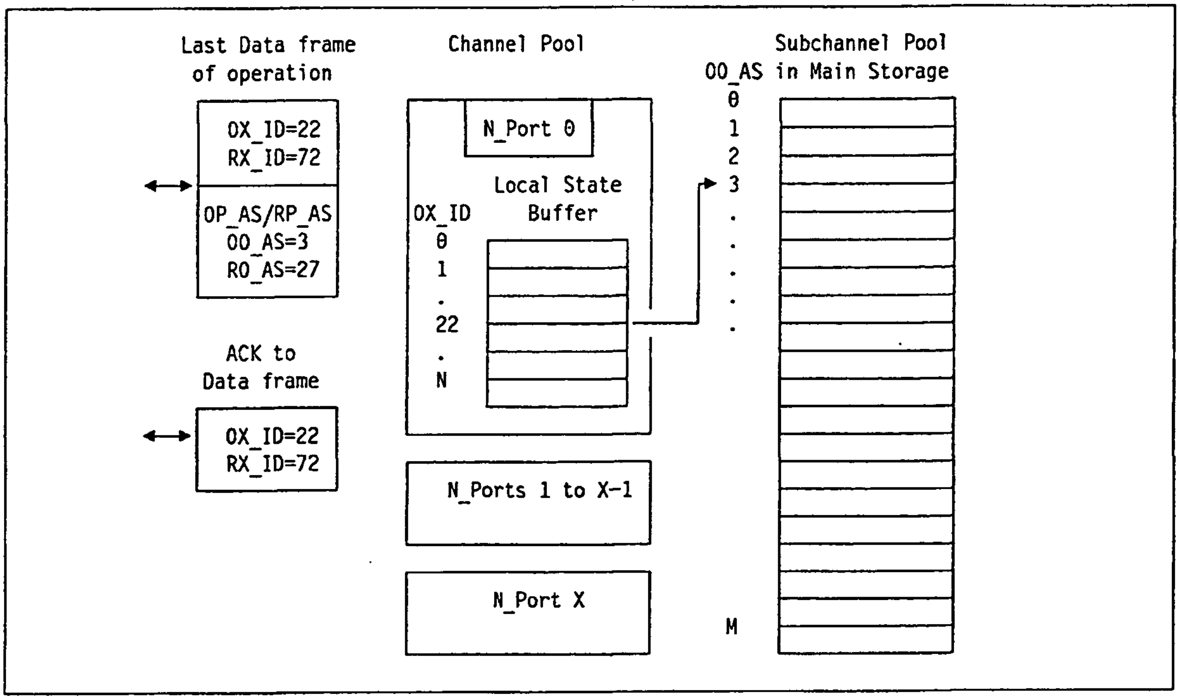


Figure R.5 - Operation termination

R.2 Process_Associator example

The Process_Associator provides an assist for routing an Exchange to a process or group of related processes at a level above FC-PH This annex describes system images in an ES/9000[9] system and provides an example of how the Process_Associator is used to route to the proper system image. The Process_Associator is mapped as a system image identifier or Partition ID.

R.2.1 Background

The physical hardware platform of an ES/9000 system can be logically configured and operated as multiple logical systems, each of which is running its own operating system, as illustrated in figure R.6. Each logical system is called a "system image". Functionally, this is identical to a configuration where each operating system resides in its own, physically separate hardware platform as illustrated in figure R.7. Logically, these two figures are identical.

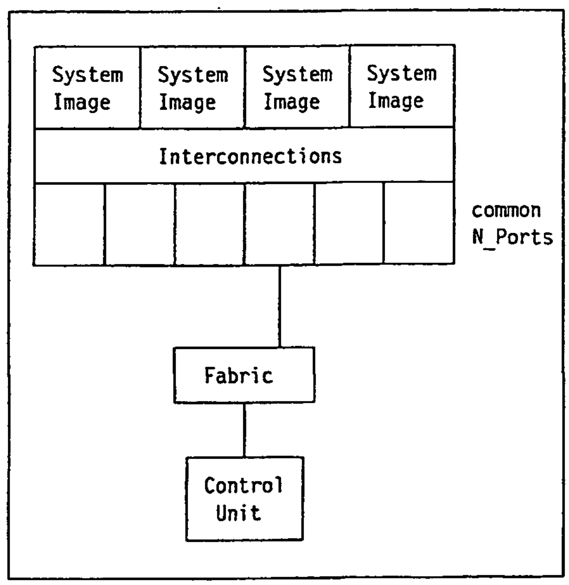


Figure R.6 - Logical System Images

Physically, the difference is that in figure R.7, the N_Port ID is sufficient to route to a particular system image. While in figure R.6, the N_Port ID only routes to an N_Port that is shared amongst

[9] ES/9000 is a trademark of the International Business Machines Corporation.

all the system images. Therefore, additional addressing is required, which is provided by the Process_Associator. This allows the N_Port to route incoming data to the proper system image. Note that the Process_Associator is simply an extension to the N_Port ID for routing.

R.2.2 Usage

When an operation is initiated to a control unit by a system image, the following occurs:

a) An Association_Header is initialized:

- The Originator Process Associator is set to indicate the system image originating the operation.
- If the control unit indicated during Login that it required an Initial Process_Associator, the channel obtains the control unit's Initial Process_Associator by a means not defined in this standard, e.g., through use of a Directory Server, or from an internal configuration table and sets the Responder Process_ Associator to this value. Otherwise, the Responder Process_Associator is set to nulls.

  Note: Some control units may have multiple system images.
- The Originator and Responder Operation_Associators are set as indicated in R.1

b) The first Data frame of the operation is transmitted to the control unit and the channel waits for the resultant ACK before sending any subsequent frames (which will not contain the Association_Header).

c) When the control unit receives this Data frame, it will:

- Remember the Originator Process_Associator and associate it with the path from which this Exchange was originated. This allows the control unit to identify any other operations from the same system image, e.g., a reset from this system image will not affect any operations to from any other syst m image.
- Generate an RX_ID and return it in the ACK to this Data frame.

When an operation to a system image is initiated by a control unit, the following occurs:

a) An Association_Header is initialized:

- The Originator Process_Associator is set to indicate the control unit image (if any) originating the operation.
- Since, on an ES/9000 system the channel indicates during Login that it requires an Initial Process_Associator, the control unit sets the Responder Process_Associator as follows:
  - If this system image had previously originated an Exchange to the control unit for the originating device, the control unit uses this "remembered" Process_ Associator as the Responder Process_Associator.
  - If the system image had not previously originated an Exchange to the control unit, for the originating device, the control unit determines the system image's Initial Process_Associator by a means not defined in this standard, e.g., through use of a Name Server, or from an internal configuration table. The control unit puts this Initial Process Associator into the Responder Process_Associator field.
- The Originator and Responder Operation_Associators are set as indicated in R.1.

b) The first Data frame of the operation is transmitted to the channel and the control unit for the resultant ACK before sending any subsequent frames (which will not contain the Association_Header).

c) When the channel receives this Data frame, it will:

- Use the Responder Process_Associator to route the incoming data for this Exchange to the proper system image.
- Generate an RX_ID and return it in the ACK to this Data frame.

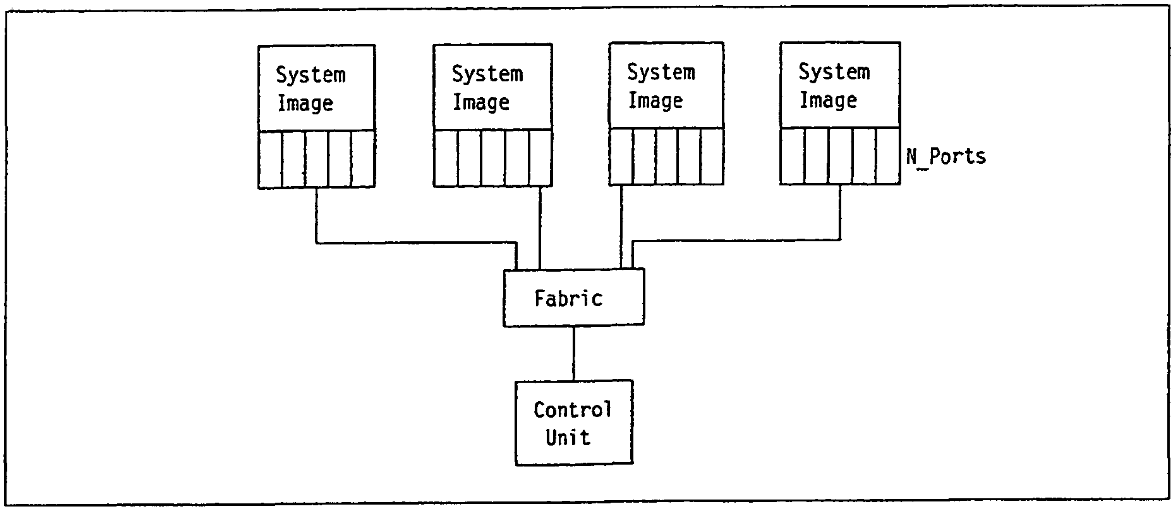


Figure R.7 - Physical System Images

Annex S
(informative)
FC-PH service interface

This annex specifies the services provided by FC-PH and the services required by FC-PH The intent is to specify the services available to FC-3 and FC-4s for the the support of Upper Level Protocols with FC-PH. How many of the services described in this annex are chosen for a given implementation is up to that implementer; however, a set of FC-PH services will be supplied sufficient to satisfy the Upper Level Protocol(s) being used. The services as defined in this standard do not imply any particular implementation, or any interface. Services described are:

a) FC-PH services provided to the local FC-4 (indicated by FC_PH_ prefix) for support of Upper Level Protocol data transfer.
b) Login Services provided by FC-PH for Fabric and N_Port Login.
c) Link Services provided to the local FC-4 entities for data services management.

Link Services, such as Login, may be performed by a common process or by individual FC-4s in an implementation specific manner.

Note - Throughout this service interface, confirmation primitives are not used with Class 3 services since Acknowledgements are not sent to confirm successful delivery.

S.1 FC_PH to ULP Data Services

The following service primitives are provided by FC-PH for use by FC-3 and FC-4 to support ULP data transfer:

- FC_PH_SEQUENCE.request
- FC_PH_SEQUENCE_TAG.indication
- FC_PH_SEQUENCE.indication
- FC_PH_SEQUENCE.confirmation

In the absence of an FC-3, the transfer of an Information Unit by an FC-4 corresponds to the transfer of a Sequence by FC-PH.

An example of a typical data transfer is shown in Figure S.1.

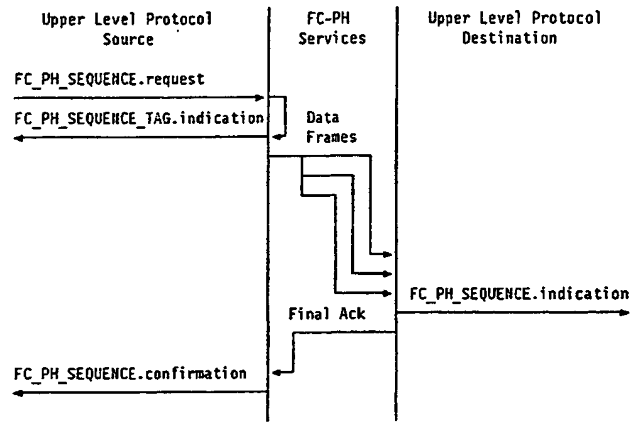


Figure S.1 - Data Transfer Service Primitives Example

S.1.1 FC_PH_SEQUENCE.request

This primitive defines the transfer of one or more Data Blocks within a single Sequence from a local ULP entity to a single peer ULP entity or to multiple peer ULP entities in the case of a group destination address.

S.1.1.1 Semantics of the Primitive

FC_PH_SEQUENCE.request
(
Type,
Exchange_Tag,
Sequence_Tag,
Allowed_Class,
Routing_Bits,
D_ID,
S_ID,
First_Sequence,
Last_Sequence,
Chained_Sequence,
Sequence_Initiative,
Continue_Sequence_Condition,
Exchange_Error_Policy,
Exchange_Reassembly,
Data_Field_Control,
Expiration_Security_Header,
Network_Header,
Association_Header,
Device_Header,
Information_Category(1),

Block_Offset(1),
Data_Block(1),
Information_Category(2),
Block_Offset(2),
Data_Block(2),
.
.
.
Information_Category(n),
Block_Offset(n),
Data_Block(n)
)

The Type will identify the Upper Level Protocol (see 18.4).

The Exchange_Tag will provide a local identifier of the Exchange that includes this Sequence. The Exchange_Tag parameter is optional on the first Sequence to allow assignment by FC-PH. The Exchange_Tag will be assigned by the ULP, the FC-4 or FC-PH to provide a unique identifier for the Exchange that includes this Sequence.

Note - The Exchange_Tag may use the Exchange_Context and Exchange_ID or any other identifier to uniquely identify the Exchange.

The Sequence_Tag may optionally be provided with this primitive. If provided in Classes 1 or 2, the Sequence Tag will be a unique identifier of the Sequence. If provided in Class 3, the Sequence_Tag shall be used by FC-PH as the Sequence ID If the Sequence_Tag is not assigned by the ULP or FC_4 and included in this primitive, it shall be assigned by FC-PH and indicated in the FC_PH_SEQUENCE _TAG.indication.

Note - This requirement implies that the FC-4 is responsible for controlling reuse of Sequence IDs by FC-PH in Class 3. If a Sequence_Tag is reused on a second IU before R_A_TOV has expired on the last frame of the previous IU, data integrity errors may result if the Sequence_Count wraps.

The Allowed_Class will indicate the set of classes of service allowed for all the frames in this Sequence (see 22). The frames for the Sequence will all be transmitted using the same class. Sequences using Class 3 will not be sent in the same Exchange with Sequences using Class 1 or 2.

The Routing_Bits will indicate the routing for all the frames in this Sequence and will indicate an allowable values specified in 18.1.2.

The D_ID will indicate the identifier of the desired destination of this Sequence (see 18.3). For Class 1 and 2, the D_ID will be an individual address; for Class 3, the D_ID may be either a individual address or a group address.

The S_ID will indicate the address identifier source of the Sequence (see 18.3.2).

The First_Sequence will indicate when the Sequence is the first in a new Exchange (see 18.5).

The Last_Sequence will indicate when the Sequence is the last in an Exchange (see 18.5).

The Chained_Sequence will indicate that a Reply Sequence is expected and will be included within this Dedicated Connection The Chained_Sequence will only be set in accordance with the conditions specified in 18.5.

The Sequence_Initiative will indicate when the Sequence Initiative is to be transferred to the Sequence Recipient on the last Data frame of this Sequence (see 18.5).

The Continue_Sequence_Condition will indicate the expected time before the next Sequence in the Exchange is presented and will take on a value specified in 18.5. The Continue_Sequence_Condition provides advisory information to FC-PH used to terminate connections and invalidate X_IDs.

The Exchange_Error_Policy will specify the error recovery policy to be used in the Exchange and is specified when the First Sequence is indicated. It will be set to a value as allowed by the Abort Sequence Condition bits in 18.5.

The Exchange_Reassembly is reserved and will be set to zero.

The Data_Field_Control will specify the complete set of optional Headers that are be included in frames of this Sequence (see 18.7)

The Expiration_Security_Header may be optionally present (see 19.2).

The Network_Header may optionally specify network addresses for the Source and Destination of the Sequence (see 19.3).

The Association_Header may optionally specify a identifier to be used to track an Exchange (see 19.4).

The Device_Header may optionally specify a Upper Level header (see 19.5).

Each set of Information_Category, Block_Offset, and Data_Block specifies a block or subblock for transmission and is referenced as a subrequest. A Data Block is a unit of data with a single Information_Category and a single associated offset.

The Information_Category may be used to provide additional control information as defined in 18.2. The number of different Information_Categories passed in a single Sequence will not exceed the value communicated by the N_Port Login Categories per Sequence parameter. The Information_Category may be the same for multiple Data_Blocks in a Sequence.

The Block_Offset may optionally specify the Relative Offset (see 18.11) for the first byte of this block or subblock of data.

The Data_Block will specify the Payload to be passed for this block or subblock of data (see 4.14.5).

S.1.1.2 When Generated

This primitive is generated by an FC-4 to request an Upper Level Protocol data transfer by FC-PH.

S.1.1.3 Effect of Receipt

Upon receipt of a FC_PH_SEQUENCE.request primitive, the source FC-PH performs the following actions

a) Receives the indicated unique Sequence_Tag or may indicate to the FC-4 a unique Sequence_Tag using the FC_PH_SEQUENCE_TAG.indication primitive.
b) Validates the parameters and verifies that the requested operation is possible, e.g., checks against login parameters, exchange initiative, etc.
c) If this is the first request for an Exchange, then enables an Exchange to be started with the first frame transmitted.
d) If this is a request that requires Exchange resources, then enables the X_ID to be assign d on th first frame of the Sequence.
e) If this is a request to release Exchange resources, then enables the X_ID to be invalidated at the end of the Sequence.
f) If this is the last request for an Exchange, then enables the Exchange to be ended with the last frame transmitted.
g) Assigns a Sequence_ID to this transfer request
h) Segments a block or subblock into frames for transmission. FC-PH determines frame boundaries in addition to those produced by the passing of multiple Data Blocks from the FC-4. If Relative Offset is supported, the Initial Relative Offset is used in the first frame of the block or subblock and the Relative Offset is computed for each succeeding frame for the block or subblock.
i) Appends an appropriate frame header and trailer to each frame.
j) Transfers the frames to the destination.
k) Maintains the Exchange Status Block and Sequence Status Blocks.
l) Uses FC_PH_SEQUENCE.confirmation to notify FC-4s as Sequences are successfully completed or abnormally terminated.

S.1.2 FC_PH_SEQUENCE_TAG .indication

This primitive provides an indication of the Exchange_Tag and Sequence_Tag for Class 1 and 2 Sequences. This primitive is optional for Class 3 Sequences.

S.1.2.1 Semantics of the Primitive

FC_PH_SEQUENCE_TAG.indication
(
Type,
Exchange_Tag,
Sequence_Tag
)

The Type will identify the Upper Level Protocol (see 18.4).

The Exchange_Tag will provide a local identifier of the Exchange that includes this Sequence. The Exchange_Tag may be assigned by the ULP, FC-4 or FC-PH to provide a unique identifier for the Exchange that includes this Sequence.

Note - The Exchange_Tag may use the Exchange_Context and Exchange_ID to uniquely identify the Exchange.

The Sequence_Tag may provide a unique identifier of the Sequence. The Sequence_Tag may be assigned by the ULP, FC-4 or FC-PH to provide a unique identifier for the Sequence.

S.1.2.2 When Generated

This primitive is atomically generated in response to the FC_PH_SEQUENCE request primitive

S.1.2.3 Effect of Receipt

The effect of receipt of this primitive by the FC-4 entity is unspecified.

S.1.3 FC_PH_SEQUENCE.indication

This primitive defines the transfer of a single Sequence of one or multiple blocks of data from FC-PH to the local Upper Level Protocol entity.

S.1.3.1 Semantics of the Primitive

```
FC_PH_SEQUENCE.indication
   (
   Type,
   Exchange_Tag,
   Class,
   Routing_Bits,
   D_ID,
   S_ID,
   First_Sequence,
   Last_Sequence,
   Chained_Sequence,
   Sequence_Initiative,
   Continue_Sequence_Condition,
   Exchange_Error_Policy,
   Exchange_Reassembly,
   Data_Field_Control,
   Expiration_Security_Header,
   Network_Header,
   Association_Header,
   Device_Header,
   Information_Category(1),
   Block_Offset(1),
   Data_Block(1),
   Information_Category(2),
   Block_Offset(2),
   Data_Block(2),
   .
   .
   .
   Information_Category(n),
   Block_Offset(n),
   Data_Block(n),
   Sequence_Valid
   )
```

The Type will identify the Upper Level Protocol (see 18.4).

The Exchange_Tag will provide a local identifier of the Exchange that includes this Sequence.

Note - The Exchange_Tag may use the Exchange_Context and Exchange_ID or any other identifier to uniquely identify the Exchange.

The Class will indicate the class of service for all the frames in this Sequence (see 22).

The Routing_Bits will indicate the routing for all the frames in this Sequence and will indicate an allowable values specified in 18.1.2.

The D_ID will indicate the identifier of the desired destination of this Sequence (see 18.3). For Class 1 and 2, the D_ID will be an individual address; for Class 3, the D_ID may be either a individual address or a group address.

The S_ID will indicate the address identifier source of the Sequence (see 18.3.2).

The First_Sequence will indicate when the Sequence is the first in a new Exchange (see 18 5).

The Last_Sequence will indicate when the Sequence is the last in an Exchange (see 18.5).

The Chained_Sequence will indicate that a Reply Sequence is expected and will be included within this Dedicated Connection The Chained_Sequence will only be set in accordance with the conditions specified in 18.5.

The Sequence_Initiative will indicate when the Sequence Initiative is to be transferred to the Sequence Recipient on the last Data frame of this Sequence (see 18.5).

The Continue_Sequence_Condition will indicate the expected time before the next Sequence in the Exchange is presented and will take on a value specified in 18.5.

The Exchange_Error_Policy will specify the error recovery policy to be used in the Exchange and is specified when the First Sequence is indi-

cated It will be set to a value as allowed by the Abort Sequence Condition bits in 18.5.

The Exchange_Reassembly is reserved and will be set to zero

The Data_Field_Control will specify the complete set of optional Headers that are be included in frames of this Sequence (see 18 7)

The Expiration_Security_Header may be optionally present (see 19.2).

The Network_Header may optionally specify network addresses for the Source and Destination of the Sequence (see 19.3).

The Association_Header may optionally specify a identifier to be used to track an Exchange (see 19.4).

The Device_Header may optionally specify a Upper Level header (see 19.5).

Each set of Information_Category, Block_Offset, and Data_Block specifies one block or subblock received and is referenced as a subrequest. A Data Block is a unit of data with a single Information_Category and a single associated offset.

The Information_Category may be used to provide additional control information as defined in 18.2. The number of different Information_Categories passed in a single Sequence will not exceed the value communicated by the N_Port Login Categories per Sequence parameter.

The Block_Offset may optionally specify the Relative Offset (see 18.11) for the first byte of this block of data.

The Data_Block will specify the Payload to be passed for this block or subblock of data (see 4.14.5).

Sequence_Valid will be set for Sequences with all frames received and valid according to the frame validity criteria in 17.8.1.

S.1.3.2 When Generat d

This primitive is generated upon the successful completion of a Sequence reception. When the Exchange_Error_Policy is not Abort_Discard_single_Sequence, within the same Exchange, the ordering of FC_PH_SEQUENCE.indications will occur in the same order as the FC_PH_SEQUENCE.requests. Upon receipt of data frames sent by the source, the destination FC-PH performs the following actions:

a) Maintains the state of the Class 1 Dedicated Connection.
b) Assembles the received block(s) and subblocks, recording any errors.
c) Maintains the Exchange Status Block and Sequence Status Blocks.
d) Issues ACK responses as required.
e) Uses the FC_PH_SEQUENCE.indication primitive to pass the completed Sequence to the appropriate FC-4.

S.1.3.3 Effect of Receipt

The effect of receipt of this primitive by the FC-4 or ULP entity is unspecified.

S.1.4 FC_PH_SEQUENCE.confirmation

This primitive will provide an appropriate response to a FC_PH_SEQUENCE.request primitive issued for Class 1 or 2 indicating the success or failure of the request.

S.1.4.1 Semantics of the Primitive

```
FC_PH_SEQUENCE.confirmation
  (
  Type,
  Exchange_Tag,
  Sequence_Tag,
  Transmission_Status,
  Reject_Reason
  )
```

The Type will identify the Upper Level Protocol (see 18.4).

The Exchange_Tag will provide a local identifier of the Exchange that includes this Sequence. The Exchange_Tag is the local identifier which was previously assigned to the Exchange.

The Sequence_Tag will provide a unique identifier of the Sequence. The Sequence_Tag is the

local Sequence identifier which was previously assigned to the Sequence.

The Transmission_Status will indicate status as one of the following:

— Successful - Sequence delivered completely to Recipient

— Unsuccessful - Sequence was not delivered completely due to abort or frame transfer error.

— Stopped_by_Recipient - Recipient stopped Sequence as indicated in ACK

— Rejected_Request - The Sequence was not sent by the Initiator due to the specified Reject_Reason.

— Rejected_by_Fabric - Reject frame received from Fabric

— Rejected_by_N_Port - Reject frame received from N_Port

When the Transmission_Status is Rejected_Request, Rejected_by_Fabric or Rejected_by_N_Port, the Reject-Reason will indicate one of the Reject reason codes given in Table 55.

S.1.4.2 When Generated

This primitive is generated upon the completion of the attempt to transmit the Sequence. This occurs after the last ACK is received indicated successful delivery of the Sequence, or after one of the Transmission_Status conditions occurs that indicates failure to deliver the Sequence.

S.1.4.3 Effect of Receipt

The effect of receipt of this primitive by the FC-4 or ULP entity is unspecified.

S.2 Login Services

Login requests are issued by a common service entity when multiple ULPs are present The following service primitives are provided by FC-PH to support Login:

— FABRIC_LOGIN.request

— FABRIC_LOGIN.indication

— FABRIC_LOGIN.confirmation

— IMPLICIT_FABRIC_LOGIN.request

— N_PORT_LOGIN.request

— N_PORT_LOGIN.indication

— N_PORT_LOGIN.response

— N_PORT_LOGIN.confirmation

— IMPLICIT_N_PORT_LOGIN.request

— N_PORT_LOGOUT.request

— N_PORT_LOGOUT.indication

— N_PORT_LOGOUT.confirmation

S.2.1 Fabric Login Primitive Flows

An example of the usage of these service primitives in a typical Fabric Login is shown in Figure S.2. Fabric Login primitive flows when no Fabric is present and a point-to-point connections exists are shown in Figure S.3.

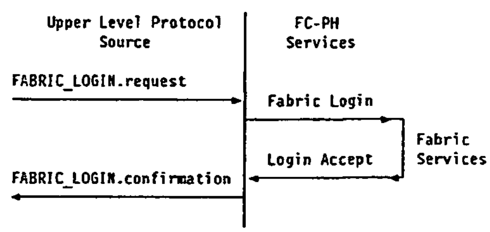


Figure S.2 - Fabric Login Service Primitives Example with a Fabric

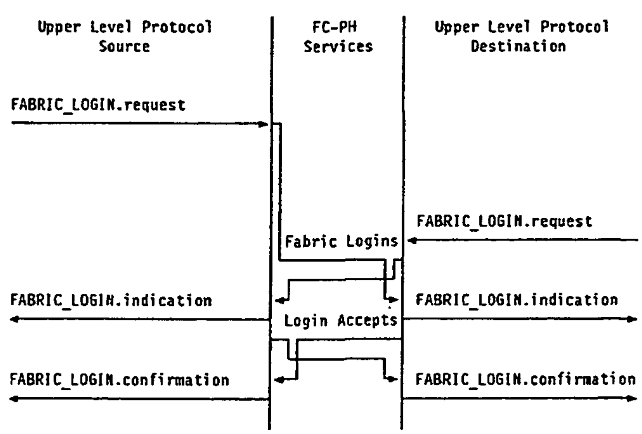


Figure S.3 - Fabric Login Service Primitives in Point-to-Point Configurati n

S.2.2 Fabric L gin Service Param ters

In the Fabric Login process, the Login Service Parameters for the N_Port are optionally provided to FC-PH with the FABRIC_LOGIN.request Service Primitive and are collectively referenced as My_Fabric_Service_Parameters. Additionally, the Login Service Parameters provided by the Fabric are optionally provided by FC-PH in the FABRIC_LOGIN.confirmation and are collectively referenced as F_Port_Fabric_Service_Parameters. Both My_Fabric_Service_Parameters and Other_Port_Fabric_Service_Parameters are of Fabric_Service_Parameters format as follows:

```
Fabric_Service_Parameters(
   fabric_common_service_parameters(
      FC_PH_Highest_Version,
      FC_PH_Lowest_Version,
      Buffer_to_Buffer_Credit,
      Buffer_to_Buffer_Receive_Data_Size,
      R_A_TOV,
      E_D_TOV
      ),
   Port_Name,
   Node/Fabric_Name,
   class_1_service_parameters(
      Class_Validity,
      Intermix,
      Stacked_Connect_Requests
      ),
   class_2_service_parameters(
      Class_Validity,
      Sequential_Delivery
      ),
   class_3_service_parameters(
      Class_Validity,
      Sequential_Delivery
      ))
```

The FC_PH_Highest_Version and FC_PH_Lowest_Version will optionally specify the supported version range (see 23.6.2.1, 23.7.1.1)

The Buffer_to_Buffer_Credit will optionally specify the buffers available as defined in 23 6.2.2 and 23.7.1.2.

The Buffer_to_Buffer_Receive_Data_Size will optionally specify the largest data frame that can be received as defined in 23.6 2.4 and 23.7.1.4.

The R_A_TOV and E_D_TOV will optionally specify appropriate timeout values as defined in 23.7.1.5 and 23.7.1.6.

The Port_Name will be specified as defined in 23.6.4 and 23.7.2.

The Node/Fabric_Name will be specified as defined in
23.6.5 and 23.7.3.

The Class_Validity will optionally be specified as per
23 6.7.1 and 23.7.4.1.

The Intermix parameter will optionally be specified as per 23.6.7.2 and 23.7.4.2.

The Stacked_Connect_Requests parameter will optionally be specified as per 23.6.7.2 and 23.7 4.2.

The Sequential_Delivery parameter will optionally be specified as per 23.6.7.2 and 23.7.4.2.

S.2.3 FABRIC_LOGIN.request

This primitive is used to provide Fabric Login parameters and to request a login with the Fabric.

S.2.3.1 Semantics of the Primitive

```
FABRIC_LOGIN.request
   (My_ID,
   My_Fabric_Service_Parameters)
```

The My_ID will specify the SID to be used in the Sequence that delivers the Fabric Login.

My_Fabric_Service_Parameters will optionally specify the parameters to be used in the payload of the Fabric Login.

S.2.3.2 When Generated

A level above FC-PH will generate this primitive to provide operating parameters to FC-PH and to request a Fabric Login.

S.2.3.3 Effect of R ceipt

The receipt of this primitive will cause FC-PH to attempt Link Initialization (see 16.6.2) if the Link is not Active and to transmit a Fabric Login Sequence with Class as specified by 23.3 3.

S.2.4 FABRIC_LOGIN.indication

This primitive is used to indicate that a Fabric Login was received from another N_Port.

S.2.4.1 Semantics of the Primitive

FABRIC_LOGIN.indication
(Other_Port_Fabric_Service_Parameters)

The Other_Port_Fabric_Service_Parameters will optionally specify the parameters from the payload of the received Fabric Login.

S.2.4.2 When Generated

This primitive will be generated upon the reception of a Fabric Login Sequence.

S.2.4.3 Effect of Receipt

Actions upon the receipt of this primitive are unspecified

S.2.5 FABRIC_LOGIN.confirmation

This primitive will provide an appropriate response to the FABRIC_LOGIN.request primitive signifying the success of the primitive and if a Fabric is present will provide the parameters returned by the Fabric.

S.2.5.1 Semantics of the Primitive

FABRIC_LOGIN.confirmation
(My_ID,
Request_Status,
Reject_Reason,
Fabric_Status,
Other_Port_Fabric_Service_Parameters)

The My_ID will reflect the D_ID returned in the Fabric Login Accept Frame.

The Request_Status will indicate status as one of the following.

— Successful - Fabric Login completed

— Unsuccessful - Sequence was not delivered completely due to reason other than reject

— Rejected_Request - The Request was not sent by the Initiator due to the specified Reject_Reason

— Rejected_by_Fabric - Reject frame received from Fabric

— Rejected_by_N_Port - Reject frame received from N_Port

— Rejected_by_Link_Services - Link Services Reject frame received from N_Port

When the Request_Status is Rejected_Request, Rejected_by_Fabric or Rejected_by_N_Port, the Reject_Reason will indicate one of the Reject reason codes given in Table 55.

The Fabric_Status will indicate status as one of the following:

— isolated - Link is not connected

— no_fabric - N_Port is connected point-to-point with another N_Port

— fabric - N_Port is connected to a Fabric

Other_Port_Fabric_Service_Parameters will optionally specify the parameters to be used for the F_Port in the operation of the Fabric when a Fabric is present as indicated by Fabric_Status, or will optionally specify the parameters to be used for the other N_Port when no_fabric is indicated by Fabric_Status.

S.2.5.2 When Generated

This primitive is generated upon the completion of a Fabric Login attempt.

S.2.5.3 Effect of Receipt

The effect of receipt of this primitive by the FC-4 entity is unspecified.

S.2.6 IMPLICIT_FABRIC_LOGIN.request

This primitive is used to perform implicit Fabric Login and provides for the specification of the complete set of Fabric parameters

S.2.6.1 Semantics of the Primitive

IMPLICIT_FABRIC_LOGIN.request
(My_ID,
Fabric_Status,
My_Fabric_Service_Parameters,
Other_Port_Fabric_Service_Parameters)

The My_ID will specify my own address.

The Fabric_Status will indicate status as one of the following:

- isolated - Link is not connected
- no_fabric - N_Port is connected point-to-point with another N_Port
- fabric - N_Port is connected to a Fabric

My_Fabric_Service_Parameters will optionally specify the parameters to be used in the operation of the N_Port.

Other_Port_Fabric_Service_Parameters will optionally specify the parameters to be used for the F_Port in the operation of the Fabric when a Fabric is present as indicated by Fabric_Status, or will optionally specify the parameters to be used for the other N_Port when no_fabric is indicated by Fabric_Status.

S.2.6.2 When Generated

FC-4 will generate this primitive to provide operating parameters to FC-PH implicitly.

S.2.6.3 Effect of Receipt

The receipt of this primitive will cause FC-PH to initialize with the specified parameters.

S.2.7 N_Port Login Primitive Flows

An example of the usage of the N_Port Login service primitives in a typical N_Port Login is shown in Figure S.4.

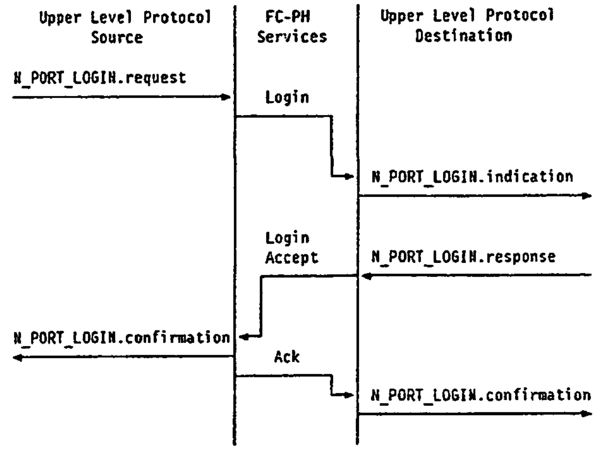


Figure S.4 - N_Port Login Service Primitives Example

S.2.8 N_Port Login Service Parameters

In the N_Port Login process, the Login Service Parameters for the N_Port are optionally provided to the originating FC-PH with the N_PORT_LOGIN.request service primitive and are collectively referenced at the originating N_Port as My_N_Port_Service_Parameters. Additionally, the Login Service Parameters provided by the responding N_Port are optionally provided by FC-PH in the FABRIC_LOGIN.confirmation and are collectively referenced at the originating N_Port as Other_N_Port_Service_Parameters.

Similarly, the Login Service Parameters passed with the N_PORT_LOGIN.response are collectively referenced at the responding N_Port as My_N_Port_Service_Parameters and the Login Service Parameters indicated with the N_PORT_LOGIN.confirmation are collectively referenced at the responding N_Port as Other_N_Port_Service_Parameters.

Both My_N_Port_Service_Parameters and Other_N_Port_Service_Parameters are of N_Port_Service_Parameters format as follows:

N_Port_Service_Parameters(
n_port_common_service_parameters(
FC_PH_Highest_Version,
FC_PH_Lowest_Version,
Buffer_to_Buffer_Credit,
Continuously_Increasing,
Random_Relative_Offset,
Buffer_to_Buffer_Receive_Data_Size,
Total_Concurrent_Sequences,

```
      Relative_Offset_by_Information_Category,
      E_D_TOV
      ),
   N_Port_Name,
   Node_Name,
   class_1_service_parameters(
      Class_Validity,
      Intermix,
      XID_Reassignment,
      Initial_Responder_Process_Associator,
      ACK_0_Initiator_Support,
      ACK_N_Initiator_Support,
      ACK_0_Recipient_Support,
      ACK_N_Recipient_Support,
      XID_Interlock,
      Error_Policy_Support,
      Categories_per_Sequence,
      Receive_Data_Field_Size,
      Concurrent_Sequences,
      N_Port_End_to_End_Credit,
      Open_Sequences_per_Exchange
      ),
   class_2_service_parameters(
      Class_Validity,
      XID_Reassignment,
      Initial_Responder_Process_Associator,
      Sequential_ACK_Transmission,
      ACK_0_Initiator_Support,
      ACK_N_Initiator_Support,
      ACK_0_Recipient_Support,
      ACK_N_Recipient_Support,
      XID_Interlock,
      Error_Policy_Support,
      Categories_per_Sequence,
      Concurrent_Sequences,
      N_Port_End_to_End_Credit,
      Open_Sequences_per_Exchange
      ),
   class_3_service_parameters(
      Class_Validity,
      Initial_Responder_Process_Associator,
      Error_Policy_Support,
      Categories_per_Sequence,
      Concurrent_Sequences,
      Open_Sequences_per_Exchange
      ))
```

The FC_PH_Highest_Version and FC_PH_Lowest_Version will optionally specify the supported version range and will equal the value specified for Fabric Login(see 23.6.2 1).

The Buffer_to_Buffer_Credit will optionally specify the buffers available point-to-point as defined in 23.6.3.2.

The Continuously_Incr asing will specify relative offset properties as defined in 23.6.3.3.

The Random_Relative_Offset will specify relative offset properties as defined in 23.6.3.3.

The Buffer_to_Buffer_Receive_Data_Size will optionally specify the largest data frame that can be received as defined in 23 6.2.4.

The Total_Concurrent_Sequences will be specified as defined in 23.6.3.4.

The Relative_Offset_by_Information_Category will be specified as defined in 23.6.3.5.

The E_D_TOV will be specified as defined in 23 6.3.6.

The N_Port_Name will be specified as defined in 23.6.4

The Node_Name will be specified as defined in 23 6.5.

The Class_Validity will optionally be specified as per
23.6.8.1.

The Intermix parameter will optionally be specified as per 23.6.8.2.

The XID_Reassignment parameter will optionally be specified as per 23.6.8.3.

The Initial_Responder_Process_Associator parameter will optionally be specified as per 23.6.8.3.

The ACK_0_Initiator_Support parameter will optionally be specified as per 23.6.8.4.

The ACK_N_Initiator_Support parameter will optionally be specified as per 23.6.8 4.

The ACK_0_Initiator_Support parameter will optionally be specified as per 23.6.8.4.

The ACK_N_Initiator_Support parameter will optionally be specified as per 23.6.8.4.

The XID_Interlock parameter will optionally be specified as per 23.6.8.4.

The Error_Policy_Support parameter will optionally be specified as per 23.6.8.4.

The Categories_per_Sequence parameter will optionally be specified as per 23.6 8.4.

The Receive_Data_Field_Size parameter will optionally be specified as per 23.6 8.5.

The Concurrent_Sequences parameter will optionally be specified as per 23.6.8.6.

The N_Port_End_to_End_Credit parameter will optionally be specified as per 23.6.8.7.

The Open_Sequences_per_Exchange parameter will optionally be specified as per 23.6.8.8.

S.2.9 N_PORT_LOGIN.request

This primitive is used to provide N_Port Login parameters and to request a login with another N_Port.

S.2.9.1 Semantics of the Primitive

N_PORT_LOGIN.request
(My_ID,
Other_ID,
My_N_Port_Service_Parameters)

The My_ID will specify the SID to be used in the Sequence that delivers the N_Port Login.

The Other_ID will specify the DID of the N_Port Login.

My_N_Port_Service_Parameters will optionally specify the parameters to be used in the payload of the N_Port Login. N_Port Login Parameters will be consistent with the parameters used in Fabric Login.

S.2.9.2 When Generated

FC-4 will generate this primitive to request an N_Port Login.

S.2.9.3 Effect of Receipt

The receipt of this primitive will cause FC-PH to transmit an N_Port Login Sequence with a Class as specified by 23.4.3.

S.2.10 N_PORT_LOGIN.indicati n

This primitive is used to provide N_Port Login parameters from a requesting N_Port.

S.2.10.1 Semantics f the Primitive

N_PORT_LOGIN.indication
(My_ID,
Other_ID,
Other_N_Port_Service_Parameters)

The My_ID will specify the DID in the received N_Port Login Sequence.

The Other_ID will specify the SID of the N_Port Login.

Other_N_Port_Service_Parameters will optionally specify the parameters that were received in the N_Port Login Sequence.

S.2.10.2 When Generated

This primitive is generated upon the receipt of an N_Port Login Sequence.

S.2.10.3 Effect of Receipt

Upon receipt of this primitive, FC-4 will optionally generate an N_PORT_LOGIN.response to cause an Accept Sequence to be sent in response to the N_Port Login.

S.2.11 N_PORT_LOGIN.response

This primitive is used to provide N_Port Login parameters for accepting a request for a Login with another N_Port.

S.2.11.1 Semantics of the Primitive

N_PORT_LOGIN.response
(My_ID,
Other_ID,
My_N_Port_Service_Parameters)

The My_ID will specify the SID to be used in the Sequence that delivers the N_Port Login Accept.

The Other_ID will specify the DID of the N_Port Login Accept.

My_N_Port_Service_Parameters will optionally specify the parameters to be used in the payload of the N_Port Login Accept. N_Port Login Parameters will be consistent with the parameters used in Fabric Login.

S.2.11.2 Wh n Generated

FC-4 will generate this primitive to respond to an N_Port Login with an Accept.

S.2.11.3 Effect of Receipt

The receipt of this primitive will cause FC-PH to transmit an N_Port Login Accept Sequence

S.2.12 N_PORT_LOGIN.confirmation

This primitive will provide an appropriate response to the N_PORT_LOGIN.request primitive signifying the success or failure of the primitive.

S.2.12.1 Semantics of the Primitive

N_PORT_LOGIN confirmation
(My_ID,
Other_ID,
Request_Status,
Reject_Reason,
Other_N_Port_Service_Parameters)

The My_ID will own address used in the N_Port Login

The Other_ID will specify the SID of the other N_port.

The Request_Status parameter will indicate status as one of the following.

— Successful - N_Port Login completed

— Unsuccessful - Sequence was not delivered completely due to reason other than reject

— Rejected_Request - The Request was not sent by the Initiator due to the specified Reject_Reason.

— Rejected_by_Fabric - Reject frame received from Fabric

— Rejected_by_N_Port - Reject frame received from N_Port

— Rejected_by_Link_Services - Link Services Reject frame received from N_Port

When the Request_Status is Rejected_Request, Rejected_by_Fabric or Rejected_by_N_Port, the Reject_Reason will indicate one of the Reject reason codes given in Table 55.

Other_N_Port_Service_Parameters will optionally specify the parameters that were received from the other N_Port during Login.

S.2.12.2 When Generated

N_PORT_LOGIN.response is generated by FC-PH in response to an N_PORT_LOGIN.request.

S.2.12.3 Effect of Receipt

The effect of receipt of this primitive by the FC-4 entity is unspecified.

S.2.13 IMPLICIT_N_PORT_LOGIN .request

This primitive is used to perform an Implicit N_Port Login.

S.2.13.1 Semantics of the Primitive

IMPLICIT_N_PORT_LOGIN.request
(My_ID,
Other_ID,
My_N_Port_Service_Parameters,
Other_N_port_Service_Parameters)

The My_ID will specify my own address.

The My_ID will specify the N_Port Login destination address.

My_N_Port_Service_Parameters will optionally specify the parameters to be used in the operation of the N_Port.

Other_N_Port_Service_Parameters will optionally specify the parameters to be used for the Other_N_Port for operation.

S.2.13.2 When Generated

This primitive is generated to provide operating parameters to FC-PH implicitly.

S.2.13.3 Effect of Receipt

The receipt of this primitive will cause FC-PH to initialize with the specified parameters.

S.2.14 N_PORT_LOGOUT.request

This primitive is used to request that a Login be terminated with the specified N_Port.

S.2.14.1 Semantics of the Primitive

N_PORT_LOGOUT.request
(My_ID,
My_N_Port_Name,
Other_ID,
Class
)

The My_ID will specify my own address.

The My_N_Port_Name is the N_Port_Name for my Port and will be specified as defined in 23.6.4.

Other_ID will specify the D_ID for the Logout Sequence.

Class will specify the Class for the Logout Sequence.

S.2.14.2 When Generated

This primitive will be generated to request a Logout Sequence.

S.2.14.3 Effect of Receipt

Receipt of this primitive will cause FC-PH to generate a Logout Sequence.

S.2.15 N_PORT_LOGOUT.indication

This primitive is used to indicate the receipt of a Logout Sequence.

S.2.15.1 Semantics of the Primitive

N_PORT_LOGOUT.indication
(Other_ID
Other_N_Port_Name,
)

Other_ID will specify the S_ID of the received Logout Sequence.

The Other_N_Port_Name is the N_Port_Name for the other Port and will be specified as defined in 23.6.4.

S.2.15.2 When Generat d

This primitive will be generated to in response to the receipt of a Logout Sequence.

Note - Logout may cause Sequences in other Active Exchanges with this N_Port to be aborted.

S.2.15.3 Effect of Receipt

The effect of receipt of this primitive is unspecified.

S.2.16 N_PORT_LOGOUT.confirmation

This primitive is used to confirm the completion of a Logout

S.2.16.1 Semantics of the Primitive

N_PORT_LOGOUT.confirmation
(Other_ID
Request_Status,
Reject_Reason
)

Other_ID will specify the D_ID for the Logout Sequence.

The Request_Status parameter will indicate status as one of the following:

— Successful - N_Port Login completed

— Unsuccessful - Sequence was not delivered completely due to reason other than reject

— Rejected_Request - The Request was not sent by the Initiator due to the specified Reject_Reason.

— Rejected_by_Fabric - Reject frame received from Fabric

— Rejected_by_N_Port - Reject frame received from N_Port

— Rejected_by_Link_Services - Link Services Reject frame received from N_Port

When the Request_Status is Rejected_Request, Rejected_by_Fabric or Rejected_by_N_Port, the Reject_Reason will indicate one of the Reject reason codes given in Table 55.

S.2.16.2 When Generated

This primitive will be generated to confirm completion of a Logout Sequence.

S.2.16.3 Effect of R c ipt

The effect of receipt of this primitive is unspecified.

S.3 Link Services

Other Link Services in addition to Login are provided. These services include:

- Abort_Exchange
- Read_Connection_Status
- Read_Exchange_Status_Block
- Read_Sequence_Status_Block
- Request_Sequence_Initiative
- Establish_Streaming
- Estimate_Credit
- Advise_Credit
- Read_Timeout_Value
- Read_Link_Status
- Echo
- Test
- Reinstate_Recovery_Qualifier
- Link_Credit_Reset
- Remove_Connection
- Abort_Sequence
- Link_Reset
- N_Port_Reset
- Link_Offline

S.3.1 LS_CONTROL.request

This primitive is used to perform Link Service actions.

S.3.1.1 Semantics of the Primitive

LS_CONTROL.request
(Control_Action,
My_ID,
Other_ID,
Request_Tag,
Target_Exchange_Tag,
Target_Sequence_Tag,
Payload)

The Control_Action param ter will include the following:

- Abort_Exchange
- Read_Connection_Status
- Read_Exchange_Status_Block
- Read_Sequence_Status_Block
- Request_Sequence_Initiative
- Establish_Streaming
- Estimate_Credit
- Advise_Credit
- Read_Timeout_Value
- Read_Link_Status
- Echo
- Test
- Reinstate_Recovery_Qualifier
- Link_Credit_Reset
- Remove_Connection
- Abort_Sequence
- Link_Reset
- N_Port_Reset

The My_ID and Other_ID optionally provide address identifiers for the request.

The Request_Tag parameter will provide a local identifier of this request.

The Target_Exchange_Tag parameter will provide a local identifier of the target Exchange when the Control_Action is Abort_Sequence, Abort_Exchange, Read_Exchange_Status_Block, or Read_Sequence_Status_Block.

The Target_Sequence_Tag parameter will provide the Sequence_Tag of the target Sequence when the Control_Action parameter is Abort_Sequence or Read_Sequence_Status_Block.

The Payload will optionally provide a payload for the requested Link Service Sequence. The payload will optionally provide additional information sufficient to identify and process the request according to Table 93.

S.3.1.2 When G nerated

This primitive is generated by FC-4 to request a Link Service.

S.3.1.3 Effect of Receipt

Upon receipt of this primitive, FC-PH issues the associated Link Service Sequence.

S.3.2 LS_CONTROL.confirmation

This primitive is used to provide a confirmation of the completion of a Link Services request.

S.3.2.1 Semantics of the Primitive

LS_CONTROL confirmation
(Control_Action,
My_ID,
Other_ID,
Request_Tag,
Target_Exchange_Tag,
Target_Sequence_Tag,
Request_Status,
Reject_Reason,
Payload)

The Control_Action is the requested Link Service.

The My_ID and Other_ID optionally are the address identifiers from the request.

The Exchange_Tag parameter will provide a local identifier of this request.

The Target_Exchange_Tag parameter will provide a local identifier of the target Exchange when the Control_Action is Abort_Sequence, Abort_Exchange, Read_Exchange_Status_Block, or Read_Sequence_Status_Block.

The Target_Sequence_Tag parameter will provide the Sequence_Tag of the target Sequence when the Control_Action parameter is Abort_Sequence or Read_Sequence_Status_Block.

The Request_Status parameter will indicate status as one of the following.

— Successful - N_Port Login completed

— Unsuccessful - Sequence was not delivered completely due to reason other than reject

— Rejected_Request - The Request was not sent by the Initiator due to the specified Reject_Reason.

— Rejected_by_Fabric - Reject frame received from Fabric

— Rejected_by_N_Port - Reject frame received from N_Port

— Rejected_by_Link_Services - Link Services Reject frame received from N_Port

When the Request_Status is Rejected_Request, Rejected_by_Fabric or Rejected_by_N_Port, the Reject_Reason will indicate one of the Reject reason codes given in Table 55.

The Payload will optionally provide the returned payload for the Link Service Accept.

S.3.2.2 When Generated

This primitive will be generated upon the completion of a Link Service request.

S.3.2.3 Effect of Receipt

The effect of receipt of this primitive by the FC-4 entity is unspecified.

Annex T
(informative)
Service interface parameters example

This annex provides an example of service interface parameters applied in conjunction with X_ID invalidation (Association_Header).

T.1 Initiate Operation

T.1.1 Parameters required from initiator upper layer

To initiate an operation, the following parameters, or information that may be used to derive the parameters, are provided to FC-PH by upper layers within the FCS node.

- Association Header (see 19.4) - the locally meaningful process and operation associators and the remotely meaningful process associator
- Local N_Port name - this may or may not be identical to the fabric-assigned local N_Port ID
- Target N_Port ID (see 18.3) - the destination N_Port ID selected by the upper layers of the FCS node at the target of the operation

These parameters are held by FC-PH for use during data transfer activities (both incoming and outgoing) associated with the operation.

T.1.2 Parameters returned by FC-PH to initiator upper layer

The following parameters are returned to the initiator upper layer by FC-PH:

- OX_ID (see 24.5) - this parameter is used by the upper layer as a reference for all subsequent activity associated with the operation.

T.1.3 Parameters received by target upper layer from recipient FC-PH

Upon receipt of the first sequence associated with a new operation, the recipient FC-PH passes the following parameters to its upper layers:

- Association Header (see 19 4) - th locally meaningful process associator and th remotely meaningful process and operation associators
- Local N_Port name - this may or may not be identical to the fabric-assigned local N_Port ID
- RX_ID (see 24.5) - this parameter is used by the upper layer as a reference for all subsequent activity associated with the operation.

T.1.4 Parameters returned by target upper layer to recipient FC-PH

- Association Header (see 19.4) - the locally meaningful process associator and operation associator (dynamically bound to this operation by the upper layer) and the remotely meaningful process and operation associators

These parameters are held by FC-PH for use during data transfer activities (both incoming and outgoing) associated with the operation.

T.2 Reconnect operation

T.2.1 Parameters required from initiator upper layer

To reconnect an operation after X_ID invalidation has occurred, the following parameters, or information that may be used to derive the parameters, are to be provided to FC-PH by upper layers within the FCS node.

- Association Header (see 19 4) - the locally and remotely meaningful process and operation associators
- Local N_Port name - this may or may not be identical to the fabric-assigned local N_Port ID
- Target N_Port ID (see 18.3) - the destination N_Port ID selected by the upper layers of the FCS node at the target of the operation

These parameters are held by FC-PH for use during data transfer activities (both incoming and outgoing) associated with the operation.

T.2.2 Parameters returned by FC-PH t initiat r upper layer

The following parameters are returned to the initiator upper layer by FC-PH:

– X_ID (see 24.5) - this parameter is used by the upper layer as a reference for all subsequent activity associated with the operation.

T.2.3 Parameters received by target upper layer from recipient FC-PH

Upon receipt of the first sequence associated with a new operation, the recipient FC-PH passes the following parameters to its upper layers:

– Association Header (see 19.4) - the locally and remotely meaningful process and operation associators

– Local N_Port name - this may or may not be identical to the fabric-assigned local N_Port ID

– X_ID (see 24.5) - this parameter is used by the upper layer as a reference for all subsequent activity associated with the operation.

T.3 Terminate operation

T.3.1 Parameters required from initiator upper layer

To terminate an operation, the following parameters, or information that may be used to derive the parameters, are to be provided to FC-PH by upper layers within the FCS node.

– Association Header (see 19.4) - the locally and remotely meaningful process and operation associators which are associated with the operation to be terminated

– Local N_Port name - this may or may not be identical to the fabric-assigned local N_Port ID

– OX_ID (see 24.5) - the OX_ID associated with the operation to be terminated.

The OX_ID previously associated with this operation is now released by FC-PH and may be reused in a subsequent operation.

T.3.2 Parameters received by target upper layer from recipient FC-PH

Upon receipt of the last sequence associated with an operation, the recipient FC-PH passes the following parameters to its upper layers:

– Association Header (see 19 4) - the locally and remotely meaningful process and operation associators which are associated with the operation to be terminated

– Local N_Port name - this may or may not be identical to the fabric-assigned local N_Port ID

– OX_ID (see 24.5) - the OX_ID associated with the operation to be terminated.

The OX_ID previously associated with this operation is now released by FC-PH and may be reused in a subsequent operation.

Annex U
(informative)
Out of order characteristics

This annex describes some of the implications of out of order transfer. There are two cases considered

a) Out of order transfer of Data frames due to the inability of a Fabric to maintain order.

b) Out of order transmission of ACKs by an N_Port due to its buffer availability algorithms.

U.1 Out of order Data frame delivery

Based on F_Port service parameters, the delivery of frames during Class 2 service may occur as

a) "Misordered Delivery". The Destination N_Port receives frames in an order different than they were sent by a Source N_Port, i.e., the Fabric does not maintain the ordering of the frames

b) "Ordered Delivery". The Destination N_Port receives frames in the same order as they were sent by the Source N_Port, i.e., the Fabric maintains the ordering of the frames.

The implications of misordered delivery and Class 2 Sequence recovery will be discussed.

Misordered frame delivery can occur whenever there are multiple routes, within the Fabric, between two communicating N_Ports. When a Sequence is initiated, the individual frames of the Sequence are independently routed by the Fabric and, therefore, may take different routes through the Fabric, with some routes being longer or shorter than others. This may cause the misordered delivery of frames to the destination N_Port Also, since each frame is independently routed, it is very difficult for the Fabric to purge, or flush from the Fabric, all the frames for a Sequence

Because of the above, this standard has provided the following functions to aid in the detection and recovery of Sequences abnormally terminated due to timeout, e.g., because a frame was lost:

a) A timeout, R_A_TOV, specified by the Fabric, which is twice the longest amount of tim that a frame may stay in the Fabric before it is either delivered, or it will never be delivered

b) Establishment of a Recovery_Qualifier range for the duration of the R_A_TOV time. If a Sequence has been aborted, the Sequence Recipient will discard any received Data frames that fall within this range, for a period of R_A_TOV. Likewise, the Sequence Initiator will discard any received Link_Control frames that fall within this range, for a period of R_A_TOV.

These functions have several implications·

a) When an N_Port is initialized, it may not have knowledge of Sequences initiated prior to initialization. For example, an N_Port may be powered off after sending a Sequence, and then powered back on. Some (or all) frames of this prior Sequence may still be traversing the Fabric after the N_Port has been initialized. Therefore, after initialization, an N_Port must wait the R_A_TOV time before it initiates any Sequences. This ensures that there will be no duplicate frames in the Fabric.

b) The establishment of Recovery_Qualifiers requires an N_Port to maintain a list of Recovery_Qualifiers. Entries must be added to this list when a Sequence is abnormally terminated, and entries must be deleted from this list when R_A_TOV has expired for the entry. The list must be referenced prior to Sequence initiation to ensure that a Data frame that falls within a Recovery_Qualifier range is not transmitted.

c) If a subset of the entire Sequence_Qualifier (e.g., X_ID) is used to route and store incoming frames, a frame falling within the Recovery_Qualifier range may not be detected until after the frame is placed in a receive buffer and the frame header is validated This has implications on credit and buffer management.

   Note that the Sequence to which this frame belongs was abnormally terminated and all the credit for the Sequence was recovered. As a result, this frame is an "unexpected" frame that is not accounted for by the current credit management within the N_Port. There-

fore, it may be occupying a buffer that a source N_Port believes is available. This may cause another frame to receive a P_BSY, even though the sender of the busied frame obeyed the credit rules. Note that this can happen to a Class 1 frame if the receiving N_Port supports intermix. Therefore, a Class 1 frame, other than SOFc1, may receive a P_BSY. in violation of the Class 1 rules.

U.2 Out of order ACK transmission

The transmission of ACK frames in either Class 1 or Class 2 service may occur as:

a) "Misordered Transmission". In this case, Data frames are not being acknowledged in the SEQ_CNT order by the Sequence Recipient. That is, the corresponding ACK frames are not being sent in SEQ_CNT order.

b) "Ordered Transmission". In this case, Data frames are being acknowledged in the SEQ_CNT order by the Sequence Recipient. That is, the corresponding ACK frames are being sent in SEQ_CNT order.

The implications of misordered transmission of ACKs and ordered transmission of ACKs are:

a) With misordered transmission, the credit for a lost ACK cannot be recovered until after a Sequence timeout is detected. That is, the credit is lost until the E_D_TOV time has expired.

b) With ordered transmission, the reception of an ACK recovers the credit for all Data frames with that SEQ_CNT or lower, regardless of whether previous ACKs were received. This is true regardless of whether the Fabric supports misordered delivery or ordered delivery.

Annex V
(informative)
Link Error Status Block

In this annex, guidelines are provided to manage the Link Error Status Block (see 29.8).

V.1 Link failure counters

Four types of Link failures are recorded in individual counters in LESB. The Link failure counters are.

- Link Failure Count (Word 0) (miscellaneous link errors)
- Loss of Synchronization Count (Word 1) (confirmed and persistent sync. loss)
- Loss of Signal Count (Word 2)
- Primitive Sequence Protocol Error Count (Word 3)

Link failure counters summary
The conditions under which individual counter increments are summarized in table V.1. (See table Q.6 and 16.4.3 for specific state changes, related nomenclature, considerations and conditions.)

V.2 Invalid Transmission Word

The Invalid Transmission Word Counter (Word 4) increments, once for every Invalid Transmission Word received (see 11.2.2.2 and 12.1.3 1), except during the following conditions:

a) No Transmission Word errors are counted if the receiver is in the Loss-of-Synchronization state (see 12.1.3)
b) No Transmission Word errors are counted if the Port is in the OLS Receive state (OL2) or the Wait for OLS state (OL3) (see 16.6.3)

V.3 Invalid CRC Count

The Invalid CRC Count (Word 5) increments, once for every received frame which meets one of the following conditions:

a) The Port is in the Active State (AC) and the received frame's CRC is in error and the frame is either missing an EOF delimiter or the EOF delimiter is an EOFn, EOFt, or EOFdt (see note under 17.6.2)
b) The Port is in the Active State (AC) and the received frame's CRC is in error (see note under 17.6.2)

NOTE - The frames received with EOFni, EOFdti, or EOFa may be excluded from consideration.

Tabl V.1 - Link Failure Counters and management

STATE	ACTIVE	LINK RECOVERY			LINK FAILURE		OFFLINE		
INPUT EVENT SUBSTATE	IDLE RECV (AC)	LR XMIT (LR1)	LR RECV (LR2)	LRR RECV (LR3)	NOS RECV (LF1)	NOS XMIT (LF2)	OLS XMIT (OL1)	OLS RECV (OL2)	WAIT OLS (OL3)
1. L > > LR	-	-	-	-	-	-	-	-	(1) Wd 3↑ (Ref.17)
2. L > > LRR	(1) Wd 3↑ (Ref.1)	-	-	-	-	-	-	-	(1) Wd 3↑ (Ref.17)
3. L> > IDLES	-	-	-	-	-	-	-	-	-
4. L > > OLS	-	-	-	-	-	-	-	-	-
5. L > > NOS	Wd 0↑ (Ref.1)	Wd 0↑ (Ref.2)	Wd 0↑ (Ref.5)	Wd 0↑ (Ref.8)	- (Ref.12)	(2) - (Ref.11)	(3) Wd 0↑ (Ref.20)	Wd 0↑ (Ref.15)	(2) - (Ref.17)
6. Loss of Signal	Wd 2↑ (Ref.21)	Wd 2↑ (Ref.4)	Wd 2↑ (Ref.7)	Wd 2↑ (Ref.10)	Wd 2↑ (Ref.13)	- (Ref.11)	- (Ref.18)	- (Ref.16)	- (Ref.17)
7. Loss of Sync > Limit	Wd 1↑ (Ref.21)	(4) Wd 1↑ (Ref.4)	(4) Wd 1↑ (Ref.7)	(4) Wd 1↑ (Ref.10)	(4) Wd 1↑ (Ref.13)	- (Ref.11)	- (Ref.18)	- (Ref.16)	- (Ref.17)
8. Event timeout (R_T_TOV)	-	Wd 0↑ (Ref.3)	Wd 0↑ (Ref.6)	Wd 0↑ (Ref.9)	Wd 0↑ (Ref.14)	-	- (Ref.19)	-	-

LEGEND

L > > means receiving from the Link,
An - entry means no change to counter

Ref.N means refer to the section listed near bottom of this table

Wd 0↑ means increment Link Failure Counter (Word 0)

Wd 1↑ means increment Loss of Synchronization Counter (Word 1)

Wd 2↑ means increment Loss of Signal Counter (Word 2)

Wd 3↑ means increment Primitive Sequence Protocol Error Counter (Word 3)

Ref.1 = 16 5 1
Ref.2 = 16.5.2.1 (a)
Ref.3 = 16 5 2 1 (b)
Ref.4 = 16 5 2.1 (c)
Ref.5 = 16.5 2.2 (a)
Ref.6 = 16.5 2 2 (b)
Ref.7 = 16.5 2.2 (c)
Ref.8 = 16.5.2.3 (a)
Ref.9 = 16.5.2.3 (b)
Ref.10 = 16.5.2.3 (c)
Ref.11 = 16.5.3.1
Ref.12 = 16.5.3.2 (a)
Ref.13 = 16.5 3.2 (b)
Ref.14 = 16.5.3.2 (c)
Ref.15 = 16.5.4.2 (a)
Ref.16 = 16.5.4.2 (b)
Ref.17 = 16.5.4.3
Ref.18 = 16.6.2 (a)
Ref.19 = 16.6.2 (b)
Ref.20 = 16.6.3
Ref.21 = 16.6.4

NOTES

1 Abnormal link responses from the attached Port.
2 A normal event if the Port is in loopback, or if the attached Port is in the Wait for OLS state.
3 Only increments if the condition occurs while performing the Online to Offline protocol.
4 This condition will not occur, since the Event Timeout occurs first.

Annex W
(informative)
Fibre Channel Bibliography

W.1 Fibre Channel standard documents

Fibre Channel standard documents are listed for reference (see figure 1).

Table W.1 - Fibre Channel Standard Documents

Document Name	FC Level	Short Description	X3T11 Project ID	Status
FC-PH	FC-0 / FC-1 / FC-2	Fibre Channel Physical Interface	755D	Approved Standard ANSI X3.230-1994
FC-PH-2	FC-0 / FC-1 / FC-2	Fibre Channel Physical Interface - 2	901D	In Development
FC-FG	Fabric	Generic Fabric Requirements	958D	In Development
FC-XS	Fabric	Cross Point Switch Fabric	959D	In Development
FC-AL	FC-2	Arbitrated Loop Topology	960D	In Development
FC-DF	Fabric	Distributed Fabric Topology	TBD	No Project
FC-IG	FC-0 / FC-1 / FC-2	Fibre Channel Implementation Guide	956D	In Development
FC-SB	FC-4	Mapping of Single Byte Command Code Sets	957D	In Development
FC-FP	FC-4	Mapping of HIPPI Framing Protocol (HIPPI-FP)	954D	In Development
FC-LE	FC-4	Link Encapsulation - Encapsulation of 802.2 like protocols	955D	In Development
SCSI-FCP	FC-4	Mapping of SCSI	X3T9.2/92-182	In Development
SCSI-GPP	FC-4	Generic Packetized Protocol	X3T9.2/91-013	In Development
FC-I3	FC-4	Revision to IPI-3 Disk Standard	496R	In Development
FC-I3	FC-4	Revision to IPI-3 Tape Standard	505R	In Development
FC-ATM	FC-4	Mapping of ATM/AAL5 Adaption Layer	To Be Assigned	In Development

W.2 Fibre Channel references

[1] A. X. Widmer and P. A. Franaszek. "A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code," *IBM Journal of Research and Development, 27*, No. 5: 440-451 (September, 1983).

[2] U S. Patent 4,486,739. Peter A. Franaszek and Albert X. Widmer. *Byte Oriented DC Balanced (0,4) 8B/10B Partitioned Block Transmission Code* (December 4, 1984).

[3] H.C LeFevre, "Single Mode Fibre Fractional Wave Devices and Polarization Controllers," Electronics Letters, Vol. 16, No. 10, pp 778-780, September 25, 1980.

Index

A

B

C

F

U

V

W

X